NOAA National Weather Service

FEWS Pilot Results

Results based on Interviews of Initial Pilot Users and the Impacts on Planning for Realization of CHPS

Sabeera Kulkarni
Micah Markman
Manuel Mattke

March 24, 2008

Table of Contents

1. Introduction

This document is the result of a series of interviews with key participants in a pilot of a basic CHPS prototype based on the Deltares Flood Early Warning System (FEWS). The goal of these interviews was to identify key issues with the FEWS that must be addressed before the system could be adopted as well as the set of most-significant risks that would need to be addressed as part of the CHPS implementation plan. This document is intended to serve as a guideline for the development of the CHPS implementation plan to ensure the plan will be viewed as a likely success by all of the key early participants in the project.

The individuals interviewed included:

- Rob Hartman, CNRFC
- Billy Olsen, ABRFC
- John Halquist, NCRFC
- Joe Intermill, NWRFC
- Christine Dietz, OHD
- Sudha Rangan, OHD
- Pedro Restrepo, OHD
- Karel Heynert, Deltaires

2. Pilot Status

The pilot of the Flood Early Warning System (FEWS) included four RFCs, and was intended to establish the basic feasibility of FEWS as a successor to the current River Forecast System operating at the NWS (NWSRFS). A team of Deltares, based in the Netherlands, deployed a modified version of FEWS to the following RFCs, and supported them during an extended test phase:

- CNRFC in Sacramento, CA
- ABRFC in Tulsa, OK
- NCRFC in Chanhassen, MN
- NWRFC in Portland, OR.

The first objective of the interviews was to baseline the overall status of what was and was not tested as part of the pilot with the intent of understanding what aspects of FEWS are least understood and should receive the most attention in the near term.

2.1 Overall Summary

With the intent of modernizing the software infrastructure for NOAA NWS River Forecast Center (RFC) forecasting operations, NWS worked to define and develop a minimally capable prototype using Deltares-FEWS. After a relatively short 4-month period, the system pilot was launched (April 2007) and a demonstration of the CHPS FEWS Pilot system was organized.

The CHPS system running at the Northwest River Forecast Center (NWRFC) and the North Central River Forecast Center (NCRFC) included the following basic functionality for purposes of the pilot:

- Unit Hydrograph model (Deltares' existing Unit Hydrograph model)
- Snow model (Deltares' existing SNOWMELT model)
- Sac-SMA with Heat Transfer model (SacSMA-HT application developed by OHD)
- Channel Routing model (Deltares' existing Muskingham model)
- Reservoir routing model (Deltares' existing reservoir model)
- Workflows that mimic existing Hydrologic Control Language (HCL) (Deltares' existing XML-based control flow)
- Basic time series transformations (Deltares' existing time-series transformation modules)
- Ability to conduct 'what-if' scenarios (Deltares' existing what-if capability)
- Estimation of missing data and computation of basin area averages from point values (Deltares' existing modules)
- Export of point data from the operational AWIPS hydrologic database to FEWS (Export Application was developed by RTi)

Based on feedback from RFCs and CAT, Deltares implemented certain additional requirements between April and December 2007, culminating in a demonstration and training workshop hosted in December 2007. A third CHPS FEWS Pilot was also installed at ABRFC in Tulsa, OK.

The additional requirements included:

- Provision of a MODS-like capability (to supplement what-if scenarios)
- Inclusion of the SNOW-17 model (to replace SNOWMELT)
- Implementation of a distributed architecture (currently standalone) to accommodate multiple forecasters working simultaneously on the same system
- Displays available in English units, not metric
- Additional training and documentation for the system
- A re-configuration of the Santiam basin at NWRFC to reflect existing NWSRFS segment definitions

Below, we describe the specific functionality evaluated at each of the pilot RFCs.

2.2 CNRFC

CNRFC did not use FEWS as an operational forecasting tool. They interacted with FEWS as the model management interface for ResSim. It served as a medium linking NWSRFS and ResSim.

2.3 ABRFC

- During the pilot installation, no calibration was required by ABRFC. Deltares ported required model parameters to equivalent models available in the pilot.
- ABRFC had a basic demonstration FEWS setup on a laptop prior to the second phase of the FEWS pilot. There was no modeling specific to their forecast area and the system used pre-loaded datasets (i.e. system was not connected to a live data feed). ABRFC had a FEWS pilot installed as a part of second round of installations and hence has had limited interaction with FEWS.
- The majority of the development or integration effort was taken care of by Deltares. Local ABRFC experts pre-configured necessary hardware and much of the software such as Postgres and Linux. NCRFC assisted by providing guidance on "standard" directory structures and data manipulation scripts. OHD installed the "ofsde equivalent" script that moved data from the AWIPS IHFS database to FEWS.
- ABRFC testers only evaluated the FEWS in single user mode

2.4 NCRFC

- During the pilot installation, no calibration was required.
- NCRFC had a small river basin modeled along with a limited hydrodynamic model for a portion of the Red River mainstem. They did not have sufficient functionality to produce a full forecast for a basin.
- Any development or integration effort was provided by Deltares and OHD. This included conversion of NWSRFS operations to comparable FEWS modules, and a rough SOBEK implementation. NCRFC provided data transfer from the IHFSDB, and additional data from NWSRFS to support SOBEK.
- NCRFC testers evaluated the system only in single user standalone mode.
- Models used in the context of FEWS at NCRFC:
    - PLOT-TS: This is not strictly PLOT-TS but rather similar functionality provided by FEWS
    - PLOT-TUL: This is not strictly PLOT-TUL but rather similar functionality provided by FEWS
    - SNOW-17 was not delivered as part of the original Pilot (standalone), but rather was delivered in December as part of the Client/Server implementation.
    - SAC-SMA
    - SAC-HT
    - SOBEK: Sobek is modeling software developed by Deltares. Sobek is not a completely calibrated model and does not usually produce results considered reasonable in the NWS' forecasting environment. The NCRFC team was able to investigate the general concepts of linked hydrologic and hydrodynamic models, and demonstrate their implementation within FEWS.
    - Ensembles were delivered to NCRFC in October 2007 in an update to the standalone system, which was then reconfigured for correct ensemble operation.

As of this writing, a complete evaluation of the ensemble system has not yet been completed.

2.5 NWRFC

- During the pilot installation, NWRFC did not perform any calibration for the pilot. They provided information about current basin configuration and historical calibration in the pilot area to RTi and Deltares.
- NWRFC had one river Santiam (part of Williamette river) modeled in part. They did not have sufficient functionality to produce a full forecast for a basin (reservoir modeling capability is a limitation).
- Any development (configuration) or integration effort was provided by Deltares and OHD. Information and data migration was provided by NWRFC.
- NWRFC testers evaluated the system in single user mode. The FEWS operator-client was installed in December but has not been tested with multiple users.
- Models used in the context of FEWS:
    - SAC-SMA
    - SNOW-17
    - Used plotting capabilities which were a part of FEWS

3. Risks Identified During Pilot Review

During the interviews, pilot participants were asked both directed and open-ended questions about the key challenges they faced during the pilot as well as the key challenges they could foresee for the actual effort to migrate onto the CHPS platform. This section serves to summarize the common and most significant challenges discussed. The challenges are listed in our interpretation of the criticality of addressing each item based on an assessment of the likelihood of the challenge impacting the project and the severity of the impact if it were to occur.

3.1 Ambiguity of Roles and Responsibilities

Up until this point of the overall project, there has been sufficient ambiguity about roles and responsibilities in the overall process that team members fear this ambiguity could impact the project as it moves forward into a more significant and complex phase. This ambiguity could lead to confusion about ownership of tasks and result either in work not getting done or getting done more than once.

The team needs to define and scope the roles and responsibilities throughout the program clearly upfront and get buy-in from everyone so resources can be planned for and the potential of key items or decisions being missed or delayed is minimized.

3.2 Changing both Operations and Science Simultaneously

The overall CHPS migration could cause significant changes to both the execution environment and the underlying science at same time. Based on the experiences from the previous migration to NWSRFS, this may be more than forecasters can reasonably handle in the desired short transition timeframes. The effort required to learn and adapt to a new mechanism for executing the forecast operations (colloquially known as the “forecasting knobology”) will be required at a minimum.

Learning the new science required to adapt to a new set of models adds extra complexity to the forecaster adoption that may not be required. While there are always improvements that can be made, the interviewed RFCs are generally satisfied with the output of their current models. Additionally, the greater work required operationalizing new models with calibration data and updates to the existing workflows may create resource scheduling bottlenecks at RFCs.

The recommended strategy is to hold one of the two variables (knobology or models) constant while migrating forecasters over on the other. Because the FEWS infrastructure provides much greater flexibility in integrating various models going forward, it is the logical choice to migrate everyone to the FEWS knobology first. Thus the team should hold a very high bar for the benefits of migrating to a new model as opposed to adapting (or even rewriting) and existing model for the initial rollout.

The cost of transitioning operations first will likely be that extra effort may need to be expended in the short term to adapt/migrate models that are not really part of future plans just to maintain the short-term status quo and simplify the transition.

3.3 No Clear Solution for Calibration

The pilot thus far has not dealt with the issue of creating and maintaining calibration data. There is also not a clear consensus plan on how to handle calibration going forward. This challenge may somewhat be addressed if all of the existing forecast operations that require calibration are

carried forward because the existing calibration data can be carried forward with the operations. However, the gap will rapidly become critical if any significant re-calibration is required or if a team wants to move to a new model.

3.4 Availability of Technical/Subject Matter Expertise

Because much of the expertise in what needs to be done and how things will need to be build resides in individuals who will still be required to do their normal jobs, there is the risk that key individuals will not be available when required.

A clear and well-defined plan with task ownership needs to be developed early in the process so people with required subject matter expertise can be scheduled in advance. There also needs to be a backup strategy in place to account for unavoidable bottlenecks (such as weather conditions or personal emergencies).

Based on task assignment, technical skill-set gap will need to be identified. The feeling among our interviewees seemed to be that RFCs themselves will need to do this analysis because no one outside their groups truly understands the staffing and skill-sets available within each RFC. These gaps will need to be addressed via training scheduled well before their tasks are set to begin work or ask that the work be shifted to another team with the required technical skills.

3.5 Perceived Performance Meeting Expectations

The existing RFS system has gone through many years of tuning to ensure the performance and user experience of the system matches exactly the needs of forecaster during high-intensity situations. Without these years of experience and tuning, the FEWS-based platform may deliver perceived performance that is not what forecasters are used to.

First off, key performance-sensitive usage scenarios need to be identified and usage expectations and metrics for the RFCs should be identified and documented.

Additionally, the existing pilot deployments should attempt to use FEWS in multi-user mode to determine how close the existing Pilot deployments are to their expectations.

Finally, a specific testing cycle should be planned during the initial system implementation to test system performance and usability during peak forecasting situations.

3.6 Total Number of “Local Applications” may affect Schedule

All RFCs have “Local Applications” external to NWSRFS and most of these applications have been developed locally by the RFCs. Each RFC has used local resources. As a result, these applications operate in diverse hardware and software environments, and are written to varying quality standards. The effort required to understand all of these applications and then migrate them to conform to the standard CHPS technologies could require significant training and work for RFCs or greater support than is available from OHD and Deltares.

The amount of “Local Application” work must be actively managed to ensure it does not overwhelm the project. This can be accomplished by focusing initial efforts only on applications that are directly dependent on an aspect of the existing infrastructure that will be changed by the migration to FEWS. Additionally, providing greater coordination capabilities could enable RFCs to share the burden of reworking the key applications that are absolutely required to maintain their operations.

3.7 Availability of Custom Technical Documentation

Since FEWS is a completely new system for NWS with potentially a new operational process as well as new models, the effective use of the system will need extensive documentation specifically reflecting the NWS environment. Thus far, there as been very good success with the standard documentation provided by the Deltares team, but this standard documentation does not and can not realistically be expected to cover all the unique specifics of NWS. To be successful, the project will require the capability for each of the participating RFCs as well as other team members to generate and share documentation with each other in an open, fluid manner, as the need to share knowledge, migration experience and local applications among the RFCs is and will be significant. Specifically, we imagine the use of one or more wikis, a source code management environment, and other options.

3.8 Simultaneous AWIPS-II and CHPS Migrations

The AWIPS-II and CHPS platform development schedules overlap, and early on, OHD recognized and accepted the fact that the development schedules of both solutions would not be synchronized. There is the significant risk that the two schedules will not mesh very well and the final AWIPS-II system will be available later than then initial RFCs are scheduled to adopt CHPS. It is likely the CHPS platform will need to be developed so that it can interface both with AWIPS and with AWIPS-II. It will be important to synchronize the overall program schedules of CHPS and AWIPS-II to ensure a seamless delivery of CHPS within the appropriate AWIPS or AWIPS-II environments.

Additionally, our interviewees expressed their perception that the AWIPS-II program includes efforts to bring more of the mission critical applications involved in the forecasting process onto a supported platform. This would likely include many of the “Local Applications” being adapted or developed during the FEWS migration efforts. The CHPS team should work with the AWIPS-II to identify any standards or requirements dictated by AWIPS-II so any “Local Applications” developed by the CHPS team can be moved to AWIPS-II with minimal effort in the future. As part of this work, certain functionality provided by “Local Applications” in the current environment may be identified for replacement by an AWIPS-II standard application (for example, data verification). Investment decisions should be made with this information in mind; for example, it may not be worth building the perfect solution to a problem if that problem is planned to be addressed by AWIPS-II in the future.

3.9 Force-fitting FEWS into Existing Hardware Environment

With FEWS being based on a software architecture different from NWSRFS, there may be different requirements for the hardware environment. Thus far in the pilot, participants have generally had sufficient success with relatively minimal hardware. However, the rigidity of the AWIPS environment combined with the lack of experience running the system in more of a distributed multi-user mode has created significant concern that the existing hardware within RFCs (both AWIPS and non-AWIPS) may not be sufficient to provide sufficient performance and availability for the new system.

Once the architectural and functional direction of the system has been established, the team will need to plan for and perform a hardware sizing exercise in which a series of stress and load tests are run against the software running on a hardware environment built based on the experiences of Deltares. Based on the results of this, a plan for hardware and any required software performance optimizations can be developed.

3.10 Lack of Documentation and Test Cases for Existing RFS Operations

Based on our discussions and previous experience with NWSRFS, it appears that many of the existing RFS Operations, particularly the most complex ones, are poorly documented and are covered by minimal or no test cases. As a result, ensuring correct functionality of adapters and migrated operation code will likely be challenging without this documentation. This deficiency will particularly make re-writing the same Operation in a new language difficult.

As a result, it may be better to adapt an existing operation even if the work may seem larger initially; especially in the case that an Operation does not really fall into the future vision for RFCs forecasting. Significant efforts will have to be made to develop a robust set of test cases that can be run against the Operation before and after adaption/migration.

3.11 Impact of Weather Conditions on Testing and Implementation Schedule

Duration for parallel operations with FEWS may be impacted by the necessity of testing the system in flood and normal situations. It is essential for RFCs to ascertain that the new FEWS system operates as expected in flood situations. Either the parallel testing schedule will need to be very flexible or there will need to be a simulation environment in which testing and training can be conducted regardless of actual weather conditions.

A flexible testing schedule will have to account for impact of variable weather conditions for each RFC and include reasonable buffers in the project schedule. This may lead to extra load on the hardware environment or extra cost to support more hardware. To manage this, it will be important to plan for a hardware environment which does not load the existing system or affect the forecasting process.

The simulation environment would need to allow forecasters to experience a variety of weather situations in a mechanism as similar to regular operations as possible. This would enable RFCs to make an accurate acceptance decision based on sufficient data in a fixed period of time. However, to also support on-the-job training needs, the environment would need to be available for long enough to support cycling in a sufficient set of the operational staff to provide expertise training to others once the system goes live.

There is the thought that the weather offices currently have a weather simulation capability. An investigation regarding the cost to take advantage of this weather simulation capability or to build up a simulation environment from scratch should be performed to decide if this additional cost is worth ensuring a limited parallel operations time frame.

Regardless of the strategy chosen for testing and parallel operations, there is a desire to keep NWSRFS up for a period of time beyond the end parallel operations for emergency fall back situations. There is the belief that it could be kept up for purely emergency situations with a fairly limited effort. This fallback procedure and the most minimal work required to keep NWSRFS in a state for this fallback scenario will need to be determined and documented.

3.12 Existing Network and Security Constraints and CHPS

The security constraints put on the NWS production environment may cause challenges for both the deployment of the system (by limiting interconnectivity between RFCs) and for post-deployment support (by limiting connectivity to production systems by offsite Deltares personnel). A formal document outlining the exact permitted and forbidden interconnectivity capabilities as

well as what impacts this would have on deployment and support options should be developed as soon as possible.

Based on this impact assessment, the CHPS leadership team can make the determination of whether changes to the security guidelines need to be escalated or whether the application trade-offs are acceptable.

Initial indications are that the deployment constraints can be resolved, particularly in the short-term, by making functionality trade-offs to deal with these constraints; the FEWS system is flexible enough to be deployed in any of the possible configurations. However, any tradeoffs must take into consideration the underlying need to pass information, particularly the output of upstream forecasts, between RFCs. For RFCs that have shared water systems, it is critical for the output of the upstream RFC to flow easily and quickly to the downstream RFC.

The post-deployment support risk will need to be addressed by further developing the diagnostic capabilities of the FEWS platform so NWS team members can be more self-sufficient and in cases of disaster provide key diagnostic information quickly and easily to Deltares.

3.13 Unique Staff Scheduling Policies and Learning Curves

Each RFC seems to have a unique set staffing and training procedures as well as very different levels of expertise currently on their teams. This operational reality may make a centralized plan for training and rollout quite difficult. The need for unique plans for each individual RFC likely will cause increased support and training costs for the overall project. A workshop to plan out and describe the roll-out process for each RFC and look for ways to find efficiencies will likely help to ensure success in the given time frames.

3.14 Budgeting and Contracting Constraints

The process of securing appropriate budget and contracts in place can be time consuming and may cause unnecessary delays. An overall project plan will need to be developed working backwards from desired release dates and it must clearly identify when contracts must be in place so the management team can work to ensure the work can start on schedule. Our recommendation would be to provide an 18-week buffer for any new contracts and 6-week buffer for add-ons to existing contracts.

4. Tasks Identified During Interviews

This section is intended to describe the key tasks that were identified during the interviews. The set of tasks listed here is focused on items that are most critical, or may be overlooked easily, and we are not attempting to provide a complete, detailed project plan. Such a plan is already in development among the various parties involved in the CHPS implementation, and will need to incorporate the items listed here.

4.1 Short-term tasks

These are tasks which need to be accomplished before the team can feel completely confident in the overall program strategy and plan and begin execution against that plan.

4.1.1 Further Develop Pilot Site Installations

While the pilot site implementations provided sufficient information to make the decision to move forward with a FEWS-based CHPS, there is remaining work that could be done with the pilot installations to further prepare for the overall migration effort.

- **Test Multi-User Mode: the multi-user client server mode of FEWS has not been very thoroughly tested yet and should be tested in greater detail to understand performance and deployment implications.**

- **Continue efforts to get Pilots RFCs to the point of being able to run a complete basin forecast in parallel with other activities as it is possible. These efforts should focus on enabling RFCs to being performing actual objective verification of the system by comparing results from the pilot install to RFS results. Additionally, pilot user should focus on ensuring the system handles normal operational needs, particularly the ability to divide up the forecasting process into multiple parallel streams of work performed by different forecasters.**

- **If CNRFC is going to be part of the initial rollout, decide whether a pilot deployment with a real basin forecast be deployed so they can have a place to begin learning skills that will eventually be required during development and migration**

4.1.2 Finalize the process that will be used to manage the program moving forward

- **Finalize definition of key roles and responsibilities**

  To address the concerns related to ambiguity of roles and responsibilities, the CHPS implementation leadership should clearly articulate the key roles and responsibilities required for successful program leadership. The following is a summary of key roles and responsibilities identified during the pilot interview process:

  - Program Project Manager(s): Responsible for the following:
    - Ensuring everyone understands their task responsibilities and when these tasks must be completed, this may only involve ensuring each team leader

understands their team's responsibilities and leaving internal team management up to that team leader.

- Managing the execution of the common team members such as OHD developers as well as contractors.
- Identifying and managing the interdependencies between work tasks
- Developing and executing against the risk management plan for the program.

- RFC Project Sponsor: ultimate decision maker at RFC also responsible for ensuring RFC is able to deliver necessary resources to ensure project success.
- RFC Project Lead: day-to-day tactical lead who can provide RFC needs to project team and manage tasks assigned to an RFC
- Deltares project lead: responsible for ensuring the FEWS product delivers required functionality and that the other team members have the necessary technical support on FEWS to succeed.

- **Define process for making decisions and get buy-in from all groups impacted by the migration.**

  To minimize the risk of unnecessary delays or team stress caused by the distributed decision making environment in which the program will need to operate, a fully detailed plan for managing the project going forward should be documented and agreed to by all key parties that are impacted by the project. First and foremost, a process for assigning and communicating responsibility for task execution needs to be defined and agreed to. Each of the RFCs interviewed is ready and willing to commit significant resources to the migration effort; however, there is fear that not having a solid process of dividing the work could lead to significant inefficiency and duplication of efforts.

  This plan needs to address how the team will decide what decisions apply to all parties and thus must be made centrally as well as the process for making these central decisions (who should have input on the decisions and who will ultimately make the decision).

  A challenging decision that will have to made frequently is what particular functionality will be required at what point in time in the implementation sequence; this decision process will be much less complicated if the first step is to develop a clear set of criteria to determine what must delivered before the initial rollout, must be delivered before the full rollout, or can be deferred to a follow-on release.

  Finally, the project team should come to a common understanding on the prioritization of remaining on schedule versus scope versus budget. Having this common understanding at the outset of the project ensures each decision-maker within the project team has a common strategy to dealing with challenges the project may face going forward.

- **Determine, document, and finalize a process for documenting and getting sign-off on software requirements.**

  To ensure the success and quality of any new software development in the environment that will likely be very distributed and geographically dispersed, a solid process for the documentation and review of requirements is absolutely critical. This process should take into consideration feedback from the teams that will do much of the software development itself, feedback from the teams that will test and train future users of the

system, and feedback from the key subject matter experts who will have to agree to the specifications.

Finally, there is the reality that large amounts of the existing NWSRFS system are not well documented and the overall effort to completely document all requirements perfectly will not be feasible given schedule and budget constraints. As a result, it will be important to find a good balance between developing sufficient documentation to enable success for the development and testing teams and bogging down the project with too much documentation.

- **Determine, document, and finalize a risk management process.**

  Any project of this size and complexity will experience challenges during its lifecycle. A distinguishing factor in satisfaction with prior related projects among RFCs was the project team's ability to anticipate and plan for these challenges. Formalizing this process will increase the likelihood of providing this level of service.

4.1.3 Support collaborative learning and development across RFCs

A common theme that developed across the interviews was the criticality of a mechanism to provide more in-depth documentation and learnings to the distributed RFCs as well as allowing them to contribute their own learnings back to the knowledge pool. There was also a general recognition that there is great potential to reduce the amount of duplicative development across RFCs. However, this has to be balanced to support the reality that there will continue to be differences between the centers and each center will need to be able to make their own updates to the software. To support this collaborative development, a technology platform should be identified that provides the following functionality:

- Enable team members to post documentation as well as collaboratively edit and review key findings; some kind of Wiki software is likely the best vehicle for this.
- Provide an issue management system that can be used to track, communicate, and assign ownership of issues as well as risks. Ideally this system can be used for defect tracking during the actual development cycle.
- Sharing source code and allow for group development against the source code.

There have been discussions started by NCRFC about bringing some form of SourceForge.org to address many of these needs that have already been identified prior to this project. The immediate next step for this effort would be to review the status of these discussions in more detail, understand whether this solution would meet the needs of the CHPS program, and, if so, what level of central support would be required to roll this out more generally.

4.1.4 Finalize High Level Scope for Initial Rollout to First Phase of RFCs

- **Finalize RFCs included in initial rollout**

  Given the complexity of this project, the initial group of RFCs will need to make a full commitment to this project, including identification of exactly who at each RFC will play the key roles defined as part of the Program Management Process. The finalized list of initial RFCs will serve to clarify the scope of software development for the first release as any functionality that is only required by RFCs not participating in the first rollout can be deferred.

While each of the pilot RFCs expressed enthusiasm for the project, there were subtle differences in overall risk-tolerance and ability to consume the associated change that seem to indicate some nervousness. This nervousness would most likely be alleviated with the more concrete expectations and commitment that would be described in the project management plan (see 4.1.1).

- **Define scope of Operations required for initial rollout.**

  During the pilot interview process, an initial set of required Operations from the existing environment was developed. This list needs to be further refined and turned into a set of actionable tasks that will provide the required functionality for migrating RFCs. The following steps are our recommended strategy for finalizing the Operation task list:

  - **RFCs to Decide Which Current Operations Can Be Phased Out Prior to Migration**

    During the review of the Operations, each RFC noted certain Operations that were currently in use but could be decommissioned within their RFC prior to the migrations (by replacing it with a better alternative already provided by NWSRFS). Each RFC will need to finalize their thoughts on which Operations fall into this category and determine whether they will be able to shift off of the Operation prior to the migration.

  - **Supplement Operations list with FEWS alternative operations:**

    Compare initial list of required Operations to functionality currently available in FEWS and identify operations that potentially could be replaced instead of migrated. For each operation that could be replaced, describe impact of replacement including impacts to forecast definitions, calibration, required training.

  - **Finalize post-migration list of Operations**

    Conduct in-depth review sessions at each initial RFC; this would include a determination of whether RFC would accept an available FEWS replacement instead of requiring migration of existing operation. Finally, a workshop of key representatives from each RFC should be organized with the objective of building consensus on which Operations will be migrated and which will be replaced. The output of this session must be the set of Operations that will be guaranteed to be supported in time for the initial rollout.

  - **Verify existence of any conversion operations required for new operation**

    For any operation that is replaced or significantly altered, an analysis is required to ensure that any difference in parameter data can be handled by existing operations. If the required parameter conversion cannot be handled, the newly required Operation must be identified and documented.

- **Finalize scope of changes to Databases and overall system architecture**

  One of the most important inputs into the process of determining the impact of the migration to the FEWS platform is to fully understand and document the impact of the migration of the existing set of databases or data-storage mechanisms. A clear picture of where data is currently housed and processed today along with a clear picture of how this will be changed with the migration to FEWS must be developed. This information is

critical early data to inform decisions related to migration of other parts of the overall system that depend on various combinations of the data stores.

- **Finalize plan for Pre-processors**

  An initial listing of preprocessors and the need for them was developed during the pilot interviews. The actual functionality provided by the existing pre-processors appears to be generally quite simple - thus it would likely be very easy to use (or implement) Java versions of these functions. However, another component of these pre-processors involves getting data out of data stores and storing the calculated results. Based on decisions related to what databases will and will not be part of the CHPS architecture, the decision of how to provide the pre-processing functionality can and must be made.

- **Finalize scope of development of User Interfaces**

  Some key CHPS architectural decisions must be made to determine which parts of the overall system will provide some key functionality so decisions can be made regarding what functionality FEWS user interfaces do (or do not) provide. These decisions must encompass both the long-term direction and the answer for the initial rollout. Each issue will require someone to research the potential options, make a recommendation, and a final decision from the program leadership.

  - **What system provides the UI for calibration**

    What will the overall CHPS architecture support maintaining and generating calibration data – specifically whether this will be done using FEWS, the existing RFS calibration tools, or something else? This particular question may have two answers – one for the initial rollout and a different one for the future. The reality is that FEWS platform does not have particularly good support for running calibrations. The recommended approach is to have a calibration environment separate from the operational forecasting environment; but even this solution may not work for NWS in the short term. A detailed analysis and workshop should be focused on resolving this issue.

    A secondary finding from the pilot interviews was the identification of the best practice to run calibration using the same data sources as the operational forecasts; this provides a greater accuracy than using a different source data for calibrating models that will then be run against a different data source. In general, it is our impression from the interviews and other experiences with NWS that NWS generally adheres to this best practice already.

  - **What system will provide the User Interface (UI) for Data Quality Control**

    There are still quite a few questions to resolve related to how users will perform the incoming data quality control:

    - How will the overall CHPS architecture support analysis and assimilation of observational data?
    - Will the existing UI's continue to be used or will this be supported by the FEWS user interfaces?
    - How does either option impact the efficiency of the forecast process?

- Should corrected data be pushed back into the overall AWIPS architecture to be distributed back out?

Based on the answers to these questions, the functionality required of FEWS will be clarified and the gaps between the current FEWS offering and desired functionality can be determined.

- **Other "Local" Applications**

  There is potential for significant effort to be required to migrate the variety of "Local Applications" that have been developed to fill in the gaps in NWSRFS and AWIPS combined functionality. In some cases this functionality is specific to an RFC, but in many cases it is common functionality that has been developed multiple times because each RFC has its own technical preferences and proficiencies and the organization infrastructure to share work has not existed.

  A baseline inventory of required "Local Applications" was gathered during the pilot interview process; however, the list is significantly shorter than anticipated. The recommended next step would be do perform a detailed, on-site session with each of the initial RFCs to determine the complete set of "Local Applications" that are used during forecast operations. Each "Local Application" needs to have its underlying inputs, outputs, purpose, and system dependencies defined as well.

  Finally, an effort to shrink and rationalize this list of "Local Applications" that must be migrated should be undertaken to minimize the risk that "Local Application" development will create a schedule bottleneck for the project. The first method to shrink the list (suggested by NCRFC) would be to determine if any of the applications will be unaffected by the underlying system changes and thus can be left "as is". There is a high likelihood that many of the applications will fall into this category. Secondly, the remaining applications can be grouped by functionality to see if the any of the applications can be rationalized into a smaller set of more robust common applications.

- **Tools to support migration of Calibration Data**

  The general assumption is that no calibration data migration utilities will be required because any model that requires significant calibration data will be adapted to FEWS and will be able to use the existing calibration data. Any new Operation that requires calibration data will need to have calibration data generated for it rather than converting existing calibration data. This assumption will need to be verified for each new Operation.

- **Define tools to migrate Segment Definitions**

  The pilot interviews pointed to a lack of understanding related to how all the XML segment definitions would be created for FEWS to run the existing forecasts.

  The approach recommended by both Apex and Deltares is to provide software that will automatically translate the vast majority of the existing HCL segment definitions to XML for FEWS. This transformation software will provide significant value as it will eliminate or reduce manual and error-prone work. There are likely to be a subset of conditions that will not translate easily with an automated system, but these can be explicitly identified during the development process and provide guidance to the RFCs for where to focus their attention. The detailed specifications of this transformation will come out decisions

related to Operation migration and can be deferred to software development. The key decision will be to come to agreement on the need for this task and then update the over project plan based on the decision.

4.1.5 Synchronize plan with AWIPS-II team

As AWIPS-II integration has been identified as a significant risk, it will be critical to synchronize with the AWIPS-II program. This includes both schedule synchronization as well as technical standards sharing to ensure any work developed by the CHPS team can be deployed to AWIPS-II in the future with minimal additional work.

4.1.6 Plan for FEWS to deployed for researchers at OHD

Ability to develop experimental models without impacting production models

Need a protocol for submitting enhancements (based on research) to FEWS

4.2 Medium-term

These are tasks identified during the interviews which need to be accomplished before the first rollout to the initial set of RFCs can complete.

4.2.1 Preparation of development environment

- **Deployment of collaborative development and knowledge sharing platform as planned earlier**

- **Selection and rollout of standardized development environment**

  This would need to include tools to support the following activities that work in the development environment chosen as standard (most likely Linux although Windows could be an option):

  - XML Editing (NWS standard XML editor)
  - Java development
  - Development against a database
  - FEWS deployment for development

4.2.2 Software development requirements discovery

- **Detailed analysis of Operations**

  Based on the set of Operations selected during the high-level scoping, a detailed analysis will need to be performed to specify exactly what will be built for each Operation. The initial indications for interviewees indicate that many of the Operations in use today or not

particularly well documented or understood and do not have good test conditions. This will need to be resolved before development can begin in earnest.

- **Detailed technical analysis of each existing Operation**

  For each Operation that is going to be adapted/migrated, a detailed technical analysis will have to be undertaken to determine first whether the Operation should be adapted or migrated to a new language based on the technical dependencies to the existing infrastructure that will be replaced. Regardless of this decision, some basic test cases will need to be developed to enable the developers to ensure they are preceding correction. If the model is going to be migrated to a new language (Java), then a much more detailed analysis will need to be undertaken to understand exactly what the specifications of the Operation is.

- **Develop list of known issues with existing Operations**

  Because migration of an Operation is likely to require significant reworking of the Operation, it is an excellent opportunity to resolve outstanding issues. For each Operation on the migration list, a listing of outstanding issues with the Operation should be gathered from all RFCs (including those not involved in the initial roll-out). The development team responsible for each Operation will need to make an assessment of the complexity of the requested modifications on which a determination of the value of making a given change can be made. The project leadership team can use this data to choose to change the specifications of the existing Operations as part of the migration.

- **Develop basic test cases for new Operations**

  For any Operation that is a new Operation not currently in use, a set of test cases based on the existing Operation that is being replaced will need to be developed to enable the development team to ensure the new Operation is meeting at least some basic expectations.

- **Detailed Gap Analysis and specifications between FEWS UI's and the RFS UI's that will be replaced**

  While there has been some analysis of the existing FEWS operational user interfaces that led to the existing extensions to support MODs, there is general agreement that further refinement of these UI's. In particular, it appears to us that NWSRFS went through many iterations of development before it was truly acceptable and that the existing RFC operational processes and NWSRFS UI's are tuned together to handle worst-case flood-type situations.

  There is the fear that without some significant tuning of the existing FEWS UI's, forecasters would not be able to execute at the speed required during peak periods. A detailed analysis of forecaster usage patterns of the existing tools in peak period should be completed to identify any key usability and performance requirements. This should be followed by a review of the existing FEWS capabilities to identify a prioritized list of enhancements including which ones would be required before the initial rollout. Provided with the list of enhancements should be a set of UI mockups that describe how users would accomplish key tasks using the new system. This set of mockups should be reviewed by key users to ensure they will truly fulfill operational requirements.

As part of this analysis, a minimum set of performance requirements need to be developed for performance testing purposes.

- **Detailed specifications of "Local Applications"**

  For each of the "Local Applications" identified as requiring a migration effort, a detailed specification and rationalization process will need to be undertaken in a more formal method to ensure each potential RFC user understands and provides feedback to exactly what the local application will do and how configurable and customizable it will be. This more formal process will most likely not be required for "Local Applications" that remain completely local and unique to a RFC.

- **Detailed analysis of diagnostic tool requirements**

  The Deltares team has a reasonably good idea of what functionality added to FEWS will make the biggest gains for ability of forecast operations to diagnose problems remotely or transfer knowledge to Deltares for off-site diagnosis. The Deltares team will need to document their thoughts on this and present it to the appropriate NWS staff to ensure everyone feels comfortable with the support plan and technology available to support the plan.

- **Detailed definition of standardized integration points**

  Each RFC has a significant need for pulling data out of the overall forecast process and providing it to other data consumers (both systems and individuals) both within NOAA and outside of NOAA.

  To improve on the overall efficiency of development, a set of standardized data integration points should be developed with the hope that these standard data can fulfill 80% of the needs of data consumers. An effort to detail out the data output needs for each of the pilot RFCs plus a rationalization process to find a set of data integration points that will provide at least 80% of required data will likely save each individual RFC significant work to develop their own individual data providers.

- **Detailed gap analysis of FEWS ensemble forecasting support**

  Based on investigations to date, the FEWS platform already provides the vast majority of required functionality to support existing ensemble forecasting processes in operational use at NWS. However, the level of investigation has not been fully sufficient to be sure that no updates will be required (and, in fact, there are indications of a couple of minor issues to date). A detailed pilot focused on operationalizing ensemble forecasting using FEWS should be implemented to get a final gap listing of FEWS ensemble capabilities.

4.2.3 Hardware analysis

Based on the existing experience of Deltares with FEWS and the experience of NWS with existing hardware, a determination will need to be made regarding what hardware the FEWS will run on. This plan will need to be synchronized with AWIPS-II plan as this strategy may change as AWIPS-II progresses.

4.2.4 Performance Test Workshop

The overall testing effort should include a simulation of flood conditions with forecasters attempting to use the new platform in most intense situation. This should be done as early in the process as possible so the overall performance of the application can be evaluated and any software or hardware performance issues can be addressed.

4.2.5 Plan and phase-in process change, training and materials

Developing all of the new process documentation and training materials for forecasters using the new system will be a significant effort that will need to be planned out and given appropriate attention during its development.

4.2.6 Develop a disaster recovery / failover plan

Establishing a process for enabling a backup environment in the case of an RFC failure is a requirement. However, this process needs to include the reality that having forecasters who have understanding of the models the area being forecast is critical. The general approach those interviewed would prefer would be for a team of forecasters from one RFC being able to travel to a backup RFC and still have access to all their data and systems to manage basic forecasting operations.

4.2.7 Develop a Plan or Set of Plans for Short and Intensive Testing Period

Every pilot RFC expressed a desire to minimize the amount of time spent running RFS and FEWS in parallel to test the new CHPS. To ensure a short parallel run time, risk 3.11 will need to be addressed and a significant effort to identify a potential solution for running CHPS in a simulation environment should be pursued.

4.2.8 Evaluate and Publish Lessons Learned

Based on the initial experiences of the pilot RFCs, a set of best practices and documentation for follow on RFCs should be developed. Having a session to specifically ensure the quality of this documentation should be planned for. These experiences may require updates to the scope, requirements, specifications, implementation plan for the follow on RFCs.

4.2.9 Begin Planning for features beyond NWSRFS

As the initial rollout of the new platform nears completion, the planning team will need to plan not only for the follow rollout of the remaining RFCs but also understand what improvements and next-generation features are immediately required by the pilot RFCs. It will be difficult if not impossible to freeze changes to the overall environment for the entire migration period. Thus, the pilot RFCs will need to prioritize the key improvements that are required first and foremost and decisions related to who can execute on this work will need to be made.

5. Beyond "Current-State" Considerations

This section describes the "next-generation" activities being explored by the future CHPS user community; although many of these activities are already in use at least in a limited capacity. This will provide an initial inventory of functionality that could be delivered during migration that would provide users a substantially better system, not just an equivalent system on a new technology.

1. Ensembles (probabilistic forecasting)
    i. Creating other forecasts from existing data. Consider realization of XEFS (supported by FEWS)
    ii. Multi-model ensembles (XCFS project)
2. Cluster computing – take advantage of multithreading
3. Distributed modeling – handle greater density of precipitation data. This is critical for FFlood guidance
4. Verification of forecasts – accuracy of forecast to actual
    i. Ability to do "hindcast" verify that improving calibration results in improved forecast
    ii. Identify biases at certain points.
5. Water supply forecasting – Ensemble and Statistical
6. Operational backup capabilities
7. Improving SS-SAC model
8. Water temperature modeling
9. Hydro-dynamic models
10. Use of conditional statements in FEWS to allow control of different combination of things in different scenarios
11. Continue to enhance the Published Interface schema (PI schema)
12. Data simulation
13. Forecaster toolbox: something to give more control to forecaster during operations
14. Quick statistics on peak-to-peak and cross-basin relationships
15. Long term deterministic forecasts (beyond 21 days)
16. Data set visualization opportunities to enable significantly better monitoring tools.
17. Water resource program – Results from ESP and regression techniques could possibly be produced and viewed together within FEWS. Tie together flood and water resource forecasting in one place.
18. Improve archive database procedures and connect to FEWS system. Programs and utilities within FEWS could access and analyze this data.

6. Overview of the "Common" End to End Forecast Process

This is a description of the end-to-end process a "generic" RFC goes through to generate a forecast. The purpose of this section is to give any future team-member a quick introduction to the general overall forecasting process so the details of any given issue can be understood within the larger context of getting a forecast out.

Step 1: Establish situational awareness on hydrology, meteorology, and integrity of computer systems.

The RFCs study the situation over the last 24hrs using AWIPS and a few local applications.

Step 2: Download data from SHEF and transfer data from IHFSDB to local forecasting database (FS5). Some problems that NCRFC faces during this stage are the significant time lag to transfer data from IHFSDB to NWSRFS from where models are run.

Step 3: Gather relevant data and perform quality control

During normal situations very few people (typically 1 or 2) are involved in the data processing. During flood situations multiple hydrologists (typically 2 to 7) are simultaneously involved in the process of generating a forecast. The data is divided into small chunks for quality control. Quality control process also involves interpolation for the data points which are absent.

NCRFC has 3 separate drainage systems which can run in parallel but the system does not take advantage of the ability to forecast the drainage systems concurrently.

Step 4: Hydrologists run models with pre-processed data to generate forecasts.

Most RFCs (NCRFC, ABRFC, NWRFC) have a 6 hour time step for forecasting. CNRFC has a 5 day time step. The initial forecast is sent to US Army Corps whose response may be factored into the final forecast. Models are almost exclusively run in an interactive mode.

Recently, NWRFC have developed a batch mode for summer low flow situations. This method reduces forecast time, but still requires frequent interactive forecasting techniques when forecast review produces questionable results.

Operational schedule

- On a weekday in general there are 5 day shifts for hydrologists
- Weekends and holidays will generally have fewer people (3-4) in operations and the number varies seasonally.
- When flooding occurs, general areas responsibilities remain; but more people are brought into operations and forecast load is shared.

Step 5: Output forecast products

- QPF contingency forecasts (ensemble forecasts)
- Weekly forecasts
- Monthly long range ensemble forecast
- FIC issues Flash Flood Guidance (FFG) and routine daily forecasts. Products (such as precipitation estimates, river forecasts, FFG, etc) are being fed to the web throughout the forecast process.
- An archive system is also fed with the forecast output

7. Notes from Interview with John Halquist

1. **End-to-end forecast process**
   1.1. Walk us through an end-to-end forecast process. What are some of the problems you encounter?

Data Gathering and Review:

First steps, gather data and perform quality control of data sources (less so gathering today, most data comes in an automated fashion with varying frequencies but generally flows into database directly). However, there isn't a lot of automatic quality control (range checking, etc.) so human intervention is often required to determine if data is bad/mislabeled. Variety of tools used to evaluate precipitation, radar, river, (to some extent even) reservoir, and temperature data. One person (hydromet data review) for much of the morning does little else than data processing (precipitation point and radar and temp data) to get it ready for use in model. Some of that time is used in radar processing (final phase of reviewing radar data). Another person spends similar (but less) time doing hydrological data review (river stages etc)

They model the area on 6-hour time steps so they don't deal with time interpolation issues (there is some estimation in the MAP processor if there's missing data, a station isn't reporting data as it should). Only hourly processing is done with radar so they accumulate hourly data into 6-hour time steps for the model.

Main problem is that data interactions occur with IHFSDB which is then transferred to internal db for the model (NWSRFS), this transfer can create issues of timeliness. The total turnaround time for correcting data can take much longer than desired. Handled by multiple runs of the processor on a timed basis to keep the lag as small as possible (every 10 minutes). Time between corrections to corrected data in model db is ~10 minutes.

Forecasters Run Various Models

They are a bit different from other RFCs in that they have 3 different drainages but modeling system isn't setup to take advantage of that. Hydrologic preprocessors apply to entire area and so no one can start working until preprocessor is complete for all data. It would be better if each drainage could be threaded separately from preprocessing to model run so a problem in one area wouldn't impact other areas.

Problem: Base model runs once data is available, (using local apps) they extract from the model time series that are then sent to core of engineer districts for their use in decision making. In areas where the US Army Corps of Engineers(COE) are involved, have to wait for response from COE on their plan before able to proceed. This plan data is moved into db and any forecasters dependent on this data can proceed.

Forecasters run models through IFP; once they are satisfied, their model outputs are resynchronized with main NWSRFS db.

Another set of extracts are sent to Corps of Engineers.

Data is sent to IHFSDB; forecaster is presented a hydrological display plus tabular data and the forecaster reviews this forecast (visually displayed is the previous forecast). The forecaster also views the time series data from other series; time series from the core. 2-4 time series of data displayed for a given location. Forecaster uses this to make any final

adjustments to time series forecast, push a button, and SHEF message is generated and sent to AWIPS for distribution

Flow timeseries are exchanged with surrounding RFCs at handoff points. We receive data from MBRFC with forecast flows for the Missouri River, and we provide data to LMRFC for the lower Mississippi River. The mechanisms for the handoff/exchange for ensemble data has not yet been established and needs to be developed within FEWS.

LocalDataManager (LDM) is a program used mostly to share meteorological data between research communities. Dedicated sharing mechanism allows people to subscribe to specific products. Allows digital sharing of time series data. NCRFC uses this program to share data with the COE, USGS, Universities, and to populate public web pages.

Telephone coordination occurs on an as-needed basis. Digital coordination happens 365 days/year; telephone coordination on high-flow and low-flow (particularly for mainstemrivers used for navigation) both ends of the spectrum.

Post processing
In conjunction with hydrologic forecasts, several other products are created along the way:

- *Images of observed precipitation*
- *Images of forecast precipitation*
- *Images and other products of flash flood guidance type info (3, 6, 12, hour flash flood guidance values to WFOs).*
- *Majority of these products are kicked off as part of normal forecast processing or on a scheduled CRON.*
- *Mostly done with local apps but sometimes in conjunction (radar processing national app produces some data that is then managed and distributed).*

QPF contingency forecasts (quantitative precipitation forecast) daily

Somewhat of an ensemble set of forecasts based on different possible predicted precipitations. Run on a schedule (early morning, once data is in, before any model adjustments, and after final adjustments and forecast). Done daily, 10 members of ensemble forecast (0 qpf -> 3 qpf)

Made available to weather offices via intranet and some time series posted to their dbs. Also issued to COE via digital sub (described above).

XNAV – Allows the view of a wide variety of data in spatial form data from IHFSDB and NWSRFS and model states, example:comparing soil moisture etc. look for discontinuity. It's a standalone app (data aggregator) displays data from a number of diff sources. Data may have to be prepared. Applications prepare data for input into XNAV e.g. A program that extracts snow water equiv from all basins and then the value from SNODAS system and then compare the two values (create difference field).

Spatial viewing capability – duplicate this extra visualization capability internal to FEWS system. So on the map view can have any number of representations of map data.

Monthly long range ensemble forecast (4th Thursday of the month corresponding with national release of long range climate forecasts)

90-day set of statistical products based on climatology that has occurred (based on 50-year history). Based on current model conditions plus 90 days into the future. Series of steps for the forecaster, most of it national supported software, only exception is a GUI that helps forecaster manage the pieces that have to be run (multiple programs involved) as well as a

product review step (make sure the outputs make sense). This is a computationally intensive process, but it need to run in a reasonable time to allow the forecasters the ability to QC the products and make any necessary reruns. Products are created for all forecast points (~400).

1.2. What issues would you foresee with the end-to-end forecast process implemented on FEWS given your experience with the pilot?

We need to resolve the issue on the disconnect between our main database IHFSDB and the modeling database. We are caught up with doing QC in IHFSDB and that would be problematic if we keep doing it since there is also a transfer mechanism to get it to FEWS. The point being made here relates to the fact that we have applications external to NWSRFS for processing data, such as MPE for radar data, and others for other gridded datasets. Do these applications need to change? Are adjustments needed in them to operate smoothly and seamlessly with FEWS?) How do we synch in rapid fashion. QC modification on each side i.e. QC done in IHFSDB vs QC done within FEWS. Goals to minimize # of local apps needed to support the forecast process by building them into CHPS. To the extent we can do that, a lot of issues are about moving data around and extracting data hoping to be able to incorporate into new system. Certainly there are pieces already present which do what local apps do. Have not yet got to the point where they are producing forecast using FEWS so that is an unknown as to what has to happen. His preference would be that forecast generation would be tightly coupled with modeling part of forecast process. So not something we do after the fact but something we do as a part of the process. When running IFP if there was a button which would allow you to select a timeseries which could then be used to generate the forecast. Some of forecast schemes have 20-30 forecast points then run it over again to generate product. Not sure if FEWS has this capability.

2. Operation/Model integration

2.1. Walk through list of operations/models and for each describe the following:

2.1.1. Was it part of the pilot and if so did the model used during the pilot provide reasonable enough results that it could be used for real forecasts?

2.1.2. Is the operation/model required for your forecasting?

2.1.3. Do you know of an existing FEWS model that can be substituted?

2.1.4. How complex is this model? Is it possible to isolate this model or do we need to rewrite this model?

2.1.5. When this model is migrated, how many and which scientific deficiencies would you like to change for the model?

2.2. How do the models perform in manual vs. scheduled, background forecasting mode?

2.3. How critical is it for users to modify model states, model functionality and model operation during forecasting?

2.4. How identical would a new model have to be to the reference model to be acceptable (ex 90%)?

It would be easiest for the whole country and RFCs to reproduce existing operations 1 to 1. Personal feeling is that this isn't really required. John would kind of like to take advantage of this to come out with an improved set of operations and overcome some of the existing compromises. If done well, we could come out with better simulations in FEWS than exist in NWSRFS but that implies more work is required. Unsure what the total workload comparison

would be between the two options and that clearly needs to be decided but tends to fall into category that 100% mirror is not required.

2.5. How are the models calibrated in the pilot, and how much, if any, of the calibration data from the current forecasting system can be carried over into the pilot? What types of tools will you need to migrate the calibration data?

Calibration in the new era; frankly not a ton of thought here yet. Initial thinking (practical experience with SAC-HT): with the SAC-HT you don't have to calibrate the frozen ground part of it. However, for a large portion of the area, current calibrations are based on using the old frozen ground model and thus the calibrations are warped to handle old technique; not strickly speaking a pure "sacramento" calibration. Expectation is that SAC-HT success is dependent on reviewing calibrations and removing anything done to handle old model.

Hopeful that OHD will provide ICP/MCP for SAC-HT, that will allow them to review calibrations for SAC-HT. They know they have to convert current operations/models to FEWS definitions, as part of this, have something connected/post-process to existing calibrations system that would create xml configurations needed by FEWS. Seems like this should be relatively easy to do. In initial phases, they would continue to use existing calibration system and then translate them to FEWS. Eventually need to develop some form of calibration capabilities as part of FEWS (but would sequence that in after at least an IOC if not full operation).

Karel: Clever way to hook in models into calibration system (similarly to FEWS operational system) and then translate output of calibration system to calibration inputs for FEWS.

Existing calibration system limitation: currently designed to operate on different data than is used operationally. Mean daily flow, daily precipitation, mean temperatures; different granularity (timescale) of data used for calibration than used for operations. Need to be able to use the same data for calibration as the data used for forecasting.

3. User Experience

3.1. What are the essential UIs that you need to maintain operability?

Data quality checking user interfaces: XNAV, MPE for radar processing (as well as qc for gauge data)

Local app (FTMP) for dealing with temperature data quality control but hopeful that within next several months (by next winter), transition to GFE (gridded field editor) for this.

NMAP (forecast rainfall); hoping to move to GFE for this as well.

Need to work out a mechanism for creating temperature grids and/or MAT's to hand to FEWS. Maybe could be done in FEWS but not explored as much as possible; allows view of gridded data, but doesn't allow editing.

Would like to see more of XNAV functionality within the actual FEWS GUIS (as opposed to external app).

Mods: Absolutely critical: limited experience using the mods other than the workshop, need to spend some time. Current UI is "in the ballpark"

XDAT: table perspective of db tables GUI for interacting/viewing/editing data within the local db (IHFSDB).

For generating forecasts: XSETS (common one) not used, OPIE is used; visualization tool for creating forecast product. Expectation is that this would be replaced within FEWS.

3.2. How well do FEWS displays reflect information needed for the operational forecast process?

3.3. Tell us about some custom UIs that you use. Which of these will need to be migrated before using FEWS?

4. Extensibility and configuration

4.1. Tell us about the local apps that are essential to your operation. Which existing RFS and AWIPS visualization tools are being used as part of the FEWS pilot?

4.2. What is the complexity and stability of the process used to exchange data with other applications or databases (either local apps or partners)?

4.3. Do you understand the level of complexity required to create data exports or data integration with other databases and/or applications? If so, is that level of complexity acceptable?

4.4. Do you understand the level of complexity involved in creating or integrating displays or visualization tools into FEWS? If so, is that level of complexity acceptable?

4.5. What concerns do you have about extensibility? How flexible is the pilot to adapt to process changes dictated by forecasting approach and philosophy at NWS?

Didn't really try to do any local development with it during the Pilot. Only thing that comes close to this is the work that went on to transfer data from SHEF to FEWS for data for SOBEK model. It went quick but Delft took care of it all.

5. Performance

5.1. What is the performance criteria and a reasonable time to run a forecast?

5.1.1. Perceived Application response time (rendering charts)

5.1.2. Perceived Backend processing speed(retrieving / storing data)

5.1.3. Can the FEWS environment offer satisfactory response times when used by multiple (>3) forecasters simultaneously?

5.1.4. What are the performance metrics/benchmarks that you would use to evaluate the new operational environment?

5.2. How does the performance of the pilot for conducting forecast runs compare to the existing forecast system, in terms of time and effort? This performance evaluation will cover both a stand-alone instance and a client-server instance of FEWS.

Model area running for Pilot is very small and perception at this point is that response time for little tiny area, seems significantly slower the IFP. Historically basin to basin is basically immediate response time (second / sub-second response time) to step through a workflow. Don't really have a full hydrodynamic model up and running but in general a run of FLDWAV on a section of red river currently is a minute or less in NWSRFS compared to several minutes for SOBEK in FEWS.

Forecasters are used to working through a model in interactive mode its "truly interactive", they are not waiting as they go from one point to the next. When they click next, the model returns next result immediately. This should be the target. In early days of IFP, it would be a few seconds between push of button and display refresh and that was extremely painful because they are running through several hundred steps in a day).

On the ESP side of things, computational time is around a minute or less and from what he's seen of pilot on the Buffalo FEWS ESP mode it is substantially slower than that.

6. Training and support

Not much knowledge around what this will really take. The fact that the functionality is not changing drastically will help. Can see widely different learning curve for some folks. Feeling is that much of this will be semi-intuitive. Dealing with XML will be a bit of a learning curve; but current situation is not that easy anyway.

6.1. How much training does new staff get before they begin to use NWSRFS?

Big component of basic training is in working with calibration system to get familiarity with model and parameters etc. That pretty much stays the same.

6.2. How many people can you train at a time while maintaining operations?

Could see a situation where they spin up a subset (one of 3 drainages) on FEWS but have a different drainage still on NWSRFS. Mixed mode operations to give experience in using the system.

Run a handful of people through some intensive training over a week or so then let them go off and work with system then follow through. Rotate rest of staff through training in groups.

6.3. What will be your approach for testing the system?

6.3.1. How many people involved in testing?

6.3.2. Duration for testing?

6.4. How long would you want the FEWS system to run in parallel before you feel confident?

6.4.1. Intensive and short process?

6.4.2. Slow gradual process?

Dependent on ability to Test in various situations. Happen to have it up during an active period with a real flood then it will probably go a lot quicker. On the other hand, if you activate and there's just low->medium flow and not much activity then you'll probably have to run it longer. As soon as we feel reasonably comfortable that it is performing to an acceptable level then flip the switch.

Very difficult to maintain multiple systems so pressure will be to keep it as short as possible. Need to exercise the system in some form of flooding situation at least a couple times.

By virtue of the way their climate works probably about a 6 month period.

6.5. What sort of support do you envision during migration?

6.6. How useable is the help system integrated with the pilot?

7. Roles, Responsibilities, Staffing

7.1. Who will migrate your local apps? [OHD or you?]

7.1.1. Do you have any staffing concerns regarding development of local apps [java development]?

There is clearly a lot of duplication of functions that have been independently done at various RFCs. They have an ITO (Information Technology Officer) specialized in IT issues. They've been trying to collaborate with another RFC regarding OPIE and between the two of them they came up with idea of a collaborative server. Set up a "Sourceforge" type environment to foster collaboration. Would like something like this exist/be supported centrally to provide a mechanism to collaborate on common tools.

However, because different offices have different technical expertise, similar functions are duplicated in different technologies because that is what "they know" [Java, Tcl/TK, Perl, shell scripts, Python, etc.].

Doesn't necessarily need to be OHD doing all the work but RFC

7.2. What general concerns do you have about this process?

Whole process will be different in CHPS/FEWS world than it has been in the past. Not sure how it should be different.

Efforts are on at hydrologic lab to deal less with coding and focus on prototyping things in "high-level-languages". There is an interest in getting to point where a conceptual program in Matlab (or whatever) will be handed that over to a Java programmer to turn that into an operational program/model. Maybe the engineering branch of OHD can provide that Java programming. Thinks that it is too much to expect each RFC to develop Java programming expertise capable of build independent applications. Need to shift RFC to more general conceptual world then hand that off to specialist to convert into an actual application.

7.3. What will be your team's interest in adoption of FEWS?

7.4. Do you have concerns about OHD's role in FEWS?

7.5. Do you have any concerns about Delft's role

7.5.1. Anything Delft can/cannot provide?

7.6. Do you have any concerns about Apex's role?

8. Features beyond NWSRFS

8.1. What other features or forecasting functionality beyond current state NWSRFS platform do you think will be cutting edge in the next 3 years that FEWS should be ready for (e.g. distributed modeling, ensemble forecasting)?

Related to Ensemble, need to start thinking about creating forecasts other than strictly streamflow. Have a lot of other information that doesn't necessarily get out as a forecast product: snow/water equivalent, soil moisture into the future, etc. Lots of data is generated that could be productized but isn't today. [XEFS plans detail a number of features for implementation during the next 3 years.]

8. Notes from Interview with Billy Olsen

1. End-to-end forecast process

1.1. Walk us through an end-to-end forecast process. What are some of the problems you encounter?

Visualize data for situational awareness of current conditions using local applications such as XNAV and XDat and baseline AWIPS apps such as D2D. Update any forecasts needing immediate attention. Quality control data and model inputs such as precipitation (HAS shift) and river stages (FIC shift). HAS shift prepares radar precipitation estimates and QPF. ABRFC uses local app (P3) instead of baseline AWIPS app (MPE) for radar precip. FIC obtains 4-day reservoir forecast data from Corps, edits past model runtime mods and makes a river forecast model run. FIC issues Flash Flood Guidance (FFG). FIC issues routine daily forecasts. During flood events, FIC assigns forecast tasks to other hydrologists and therefore multiple people are accessing the forecast system. Flood forecast products are issued and coordination with customers is accomplished. Products (such as precip estimates, river

forecasts, FFG, etc) are being fed to the web throughout the forecast process. An archive system is also fed in the background during the forecast process. This process is repeated on a 6-hour cycle as operations dictate. However, forecasts may be updated at anytime of the day or multiple times within the 6-hour cycle. If there is no threat of flooding, river forecasts are issued once per day. A local app, Fcst_prog, is used to monitor the status of river forecasts. Both the FIC and HAS shift continue to monitor the hydromet situation throughout their shift.

ABRFC routinely issues two types of river forecasts – forecast with/without QPF (future rainfall for 12hrs).

Issues:

- *bad data (can't quality control everything)*
- *obtaining reliable data from manual observers at non-automated river points*
- *publishing data to web on time*
- *communications issues with customers (either failure or slowness of AWIPS network)*
- *when multiple forecasters are trying to visualize (some problems with multithread) (multiple people downloading OFS file so they can look at it IFP)*
- *Performance related issues with system itself.*

1.2. What issues would you foresee with the end-to-end forecast process implemented on FEWS given your experience with the pilot?

Although we did not get the opportunity to test the multi-user system, we do have concerns with system performance with multiple users (worried about that).

There are no gridded data editing tools. These tools need to be provided and integrated into the main forecasting system (i.e., not a standalone operation).

Staff training is an issue…staff must have in depth knowledge and experience w/FEWS to make it a success.

There is no end product composition tool (for text, graphic, binary products) available. Must have tools to both issue and monitors forecast products.

We are concerned about archive capabilities.

We are concerned that there are too many mouse clicks and otherwise there are difficulties for a forecaster interacting with the system to produce a forecast. This may be due to unfamiliarity but may also be due to more of a batch forecast mode environment for FEWS versus a required model interactive environment for RFC forecasting.

2. Operation/Model integration

2.1. Walk through list of operations/models and for each describe the following: See attached spreadsheet for answers to 2.1

2.1.1. Was it part of the pilot and if so did the model used during the pilot provide reasonable enough results that it could be used for real forecasts?

2.1.2. Is the operation/model required for your forecasting?

2.1.3. Do you know of an existing FEWS model that can be substituted?

2.1.4. How complex is this model? Is it possible to isolate this model or do we need to rewrite this model?

2.1.5. When this model is migrated, how many and which scientific deficiencies would you like to change for the model?

2.2. How do the models perform in manual vs. scheduled, background forecasting mode?

During floods it is very manual / "on demand".

On a slow day the FIC will run through a more normal set of steps but they still manually trigger the models.

Every time the models run, the raw model product gets published to the web (some of the more active WFO's monitor the published data); some other customers (ex USGS) will use data to position people/resources.

2.3. How critical is it for users to modify model states, model functionality and model operation during forecasting?

Absolutely critical; unit-hydrograph mods (current models assumes uniform rain fall over 6 hours which almost never happens). Because this assumption doesn't work in nature the forecaster has to go in and change model state to handle greater

Model can crash due to bad data such as future Temperatures r observed stages and therefore have to be able to override/correct model states.

Mod List:

- *Areal extended snow cover mod*
- *Change blend mods (blend period between observed data and model state)*
- *Ignorets (ignore observed data, rating curve is off)*
- *Melt Factor Correction (change range of snow melting)*
- *Rain Snow Mod (change rain to snow and vice versa)*
- *ROCHANGE, ROMult (Change or multiply computed runoff)*

- *RRIchange, RRIMult (Change or multiply computed precipitation)*
- *SACBASEF – add or subtract lower zone flow to update SAC model*
- *SACCO – change Sac model parameters*
- *SetMissing (take bad data completely off the plot)*
- *UHGCHNG – warp the unit hydrograph to match the event*
- *WEADD/WECHNG: update water equivalents*

2.4. How identical would a new model have to be to the reference model to be acceptable (ex 90%)?

100% match is not a requirement.

Higher tolerance for difference than some.

Policy is to have people monitor system so people can update configuration of the model to improve.

2.5. How are the models calibrated in the pilot, and how much, if any, of the calibration data from the current forecasting system can be carried over into the pilot? What types of tools will you need to migrate the calibration data?

Pilot: Used SAC-SMA and they didn't port the calibrations over (Delft did that). It was done for them.

Future: If all existing models are migrated then they'll move over existing calibrations; otherwise they'll have to recalibrate. Big calibration issues are SAC-SMA, SNOW17, RES-SNGL, Unit-Hydrograph, Routing

3. User Experience

3.1. What are the essential UIs that you need to maintain operability?

xnav (data visualization)

xsets (river forecast composition tool)

P3 (radar precipitation estimation tool)

nmap (QPF generation tool)

xdat (gui to interface w/IHFS database to display tabular data including ARCHIVE data)

GIS tool (for development and operations such as FOP)

IFP (interactive forecast program)

Mods interface for managing mods

fcst_prog (monitor status of river forecasts)

sac-sma display (display state contents of model sac-sma)

shift log (enter info/maintain shift log)

gridded data editor

met model visualization (D2D)

3.2. How well do FEWS displays reflect information needed for the operational forecast process? *---Not enough experience---*

3.3. Tell us about some custom UIs that you use. Which of these will need to be migrated before using FEWS? *---see item 3.1 above---*

4. Extensibility and configuration

4.1. Tell us about the local apps that are essential to your operation. Which existing RFS and AWIPS visualization tools are being used as part of the FEWS pilot?

"Forecast Progression" (displays past forecasts and current for a configurable time period plus observed data); allows you to monitor whether forecast is badly out of date and/or QC forecast before issuing.

XNAV, XDAT for reviewing and QC'ing data See 3.1 above.

4.2. What is the complexity and stability of the process used to exchange data with other applications or databases (either local apps or partners)? *Due to complexity of customized NWSRFS data base, data must be exchanged with custom interfaces (ofsde). Due to extreme security restrictions, it is difficult to exchange data with customers/partners.*

4.3. Do you understand the level of complexity required to create data exports or data integration with other databases and/or applications? If so, is that level of complexity acceptable? *We have a low level of understanding since this was handled in the pilot primarily by DELFT.*

4.4. Do you understand the level of complexity involved in creating or integrating displays or visualization tools into FEWS? If so, is that level of complexity acceptable? *Easy for DELFT. We have no idea.*

4.5. What concerns do you have about extensibility? How flexible is the pilot to adapt to process changes dictated by forecasting approach and philosophy at NWS? *The pilot seemed to be very flexible for experts (DELFT) but we could not do changes with our limited experience.*

5. Performance

5.1. What is the performance criteria and a reasonable time to run a forecast?

5.1.1. Perceived Application response time (rendering charts) *Maybe 15 seconds to run a forecast for the entire ABRFC area.*

5.1.2. Perceived Backend processing speed(retrieving / storing data) *Maybe 5 seconds to retrieve model files ready for IFP and a few seconds to store model results*

5.1.3. Can the FEWS environment offer satisfactory response times when used by multiple (>3) forecasters simultaneously? *No experience.*

5.1.4. What are the performance metrics/benchmarks that you would use to evaluate the new operational environment? *Time to rerun an hour of radar precip estimation, time to parse/post SHEF data input products, time to run the model, time to load a workstation interactive session of the model, time to store data from an interactive model session, time to display IHFS data in tabular and graphical fashion, etc. Overall, ideally all functionality should be faster than current operations or at minimum, no slower.*

5.2. How does the performance of the pilot for conducting forecast runs compare to the existing forecast system, in terms of time and effort? This performance evaluation will cover both a stand-alone instance and a client-server instance of FEWS.

Limited experience in standalone mode seemed comparable to current state. Some conceptual issues that can cause runs to take minutes instead of under a minute (you can run a lot more historical data through it).

Didn't get to test multi-user situation and that's where the rubber really hits the road.

TODO, come up with worst case peak usage situation and what sort of performance expectations you would have in that situation.

6. Training and support

6.1. How much training does new staff get before they begin to use NWSRFS?

1on1 training + progressive on the job training. Progressive responsibility + knob-ology in 3 months. 3-6 months until they can go solo assuming necessary aptitude; wouldn't be on alone during a flood situation.

6.2. How many people can you train at a time while maintaining operations?

Get 1 to 3 people up to speed early and fast; help plan the transition. Dedicate off-time to training on new platform (if no flood going on then can spend lots of time on parallel operations/ training).

3-6 months would be optimal parallel run time.

6.3. What will be your approach for testing the system?

6.3.1. How many people involved in testing?

6.3.2. Duration for testing?

6.4. How long would you want the FEWS system to run in parallel before you feel confident? *Emphasize an intensive and short period. 3 to 6 months may be ideal but would also want a significant flood event during the parallel ops.*

6.4.1. Intensive and short process?

6.4.2. Slow gradual process?

6.5. What sort of support do you envision during migration? *We must have significant contractor support from DELFT. Also must optimize use of resources of both OHD and other RFCs in the process.*

6.6. How useable is the help system integrated with the pilot?

For Basic knobology, documentation, help, support was great.

7. Roles, Responsibilities, Staffing

7.1. Who will migrate your local apps? [OHD or you?]

7.1.1. Do you have any staffing concerns regarding development of local apps [java development]? *Yes, many.*

Like the idea of moving to a common set of local apps but this will be very difficult to accomplish and if done may significantly delay the project. Reality is that they will have to do local applications there; if there is a plan for doing some of the work in common.

Over the last 10 or 12 year, this office moved from 8 computing platforms; so they don't think it's that big a deal in itself. However, this all takes resources away from other priorities and operational activity (flooding) cannot be predicted.

Missing Java expertise in house; mixed signal around what technologies people should be trained on.

7.2. What general concerns do you have about this process?

Parallel operations: it's best to keep parallel ops as short as possible; bulge in hardware requirements.

Need to resolve hardware issues and get them through the AWIPS lag process. Don't want to continue on one size fits all AWIPS method of hardware platform.

- *Agrees with Rob Hartman network security issue.*
- *How not to lose institutional smarts/investment in existing models.*

7.3. What will be your team's interest in adoption of FEWS?

- *System maintainability; efficiency of system maintenance.*
- *Faster science to operations*
- *Integration with customers/outside users.*
- *Sharing models with other RFCs and outside users.*
- *Easier to use multiple models in operations and to test new models.*
- *Better performance and forecast accuracy.*
- *Better customer service/meet customer requirements in more timely manner.*

7.4. Do you have concerns about OHD's role in FEWS?

Worried about OHD understanding of what RFCs are doing and thus don't really understand what the resource/training needs are of the RFCs. Very mixed signals on what skill sets are required.

Lack of clarity; current model support response has been spotty. Experts have high turnover rate at OHD.

7.5. Do you have any concerns about Delft's role

7.5.1. Anything Delft can/cannot provide?

Need to be a major player in migration and need to be around for a substantial period of time afterwards to provide support.

7.6. Do you have any concerns about Apex's role?

Very little clarity on what this would be.

Feel RTI is an unneeded middleman.

8. Features beyond NWSRFS

8.1. What other features or forecasting functionality beyond current state NWSRFS platform do you think will be cutting edge in the next 3 years that FEWS should be ready for (e.g. distributed modeling, ensemble forecasting)?

Distributed modeling allows improving the model to handle greater density of precipitation data and get around unit hydrograph issues. DM is critical for gridded flash flood guidance purposes and is a big feature of future flash flood program development.

They do run the distributed model; not critical to operations but it is there and available. Can't be modified at all (it's just run every hour on a chron. Operational distributed model is used at a limited number of locations currently and is used for a "second opinion" when issuing river forecasts.

Probabilistic Forecasting: current deterministic forecasts and long term probabilistic forecasts (AHPS) needs to be supplemented with short term probabilistic forecast capabilities and deterministic forecast need to have uncertainty bounds be a part of the product.

Verification: accuracy of actual to forecast

Water supply forecasting (both ensemble and statistical forecasting)

Gridded editor is required.

Operational Backup capabilities must be addressed.

Deployment and System Management

8.2. What are your concerns about deployment related issues?

Management and resources required for deployment is a huge concern when considering we must also transition to AWIPS-II in the same time frame.

Need a xml editor to configure FEWS, AWIPS needs to provide a standard editor.

8.3. What do you do today to support NWSRFS? How will it be different for FEWS?

We maintain our own model parameters/states and request support from HQ when there are bugs in the NWSRFS or we cannot first "fix" the problems locally. This concept will probably not change with FEWS although we think it is critical that DELFT be contracted for model support in the future.

8.4. What do you think about the suitability of the current hardware environment?

Needs updated and expanded capabilities with or without FEWS in the picture.

8.5. How would client-server approach be deployed at NWS? Would you like the concept of a central server (any RFC would be able to forecast for any other RFC using central server data)?

Using a central server is an excellent idea to explore; key issue with this: unlikely to be able to have any RFC forecast for any other; but instead, RFC personnel back up themselves and may move to a secondary location and still use the existing data etc. Hydrology and customer base is too different for any RFC to double for another; at best would be "buddy-RFCs" for some limited backup support tasks.

8.6. Data Archival process

8.6.1. Describe your "Data archival process"

We maintain 12 days of basic data in the IHFS database. Data are continuously fed to our backup (AX) database as well where full resolution data in maintained back to the early 1990's and limited data back to the 1940's. We also maintain product archives. We also maintain archives on a local standalone PC and on an online web accessed archive.

8.6.2. Do you need to access archived data? In what situation?

Yes at all times…during operations, development, verification, backup, database troubleshooting and customer requests.

8.7. How does the workflow management capability in the pilot match the current power of the hydrologic control language (HCL)?

8.8. Backup capability (disaster recovery):

Southern Regions maintain a system of laptop computer backup capability, with data uploads/downloads at the SRHQ and other internet data sources. Backup can be performed with this system at our local facility, another NWS facility or any remote location with telephone and internet service.

There is also a national team underway trying to address national issue of RFC backup. This team should consider backup capabilities of the FEWS system.

9. Notes from Interview with Rob Hartman

1. End-to-end forecast process

1.1. Walk us through an end-to-end forecast process. What are some of the problems you encounter?

Historical Perspective:

1. *Pre-processing – Local Apps*
2. *Processing – NWSRFS*
3. *Post-processing – Local Apps*

There is a desire to have a more centralized approach to both pre and post processing.

Typical forecast period is 5 days

NWSRFS is extremely reliable and this reliability is a key for success.

NWSRFS has an existing set of preprocessors (that all need significant modernization). The CNRFC was not involved in the FEWS pilot that prototyped watershed simulation and forecasting (only RESSIM work) so they have limited exposure to features that FEWS offers.

Key point here: break forecast into segments and team each works

Post-processing: suite of products/services (text/graphics) generated for customers. This is likely to remain a local app; need a way to migrate the local post-processing to FEWS data source although there is potential for increased standardization.

1.2. What issues would you foresee with the end-to-end forecast process implemented on FEWS given your experience with the pilot?

Pilot implementation was significantly different than their operational model because it focused on RESSIM model. The CNRFC has not implemented watershed modeling in FEWS at this time, but we're working toward that. Current work involves using FEWS as the intermediary between NWSRFS and HEC-ResSim. This is operational.

2. Operation/Model integration

2.1. Walk through list of operations/models and for each describe the following:

2.1.1. Was it part of the pilot and if so did the model used during the pilot provide reasonable enough results that it could be used for real forecasts?

2.1.2. Is the operation/model required for your forecasting?

All arithmetic functions are required and these should not be complex.

2.1.3. Do you know of an existing FEWS model that can be substituted?

2.1.4. How complex is this model? Is it possible to isolate this model or do we need to rewrite this model?

2.1.5. When this model is migrated, how many and which scientific deficiencies would you like to change for the model?

2.2. How do the models perform in manual vs. scheduled, background forecasting mode?

- *Most runs are manual*
- *Ensemble processes and save carry over are scheduled*
- *Open to more scheduling.*

2.3. How many of the models do users modify model states, model functionality and model operation during forecasting?

Modifications to SAC, SNOW and Time Series, Forecast Rain Snow Elevation take place outside the system and are provided as an input to the model (this is non-standard and deserves some re-examination, originally done because temperature forecast from mean aerial temps were not that good, if these forecasts are better then this may not be necessary).

Other, less common mods include RIMULT, ROMULT, MFC. CHGBLEND is perhaps the most heavily used mod of all. UHGCHNG gets some use, especially when bringing a new segment online.

Don't do MODS with Res-SNGL

2.4. How identical would a new model have to be to the reference model to be acceptable (ex 90%)?

100%. The issue is that any change that affects inputs, outputs, or results would require recalibration which is a very big deal.

2.5. How are the models calibrated in the pilot, and how much, if any, of the calibration data from the current forecasting system can be carried over into the pilot? What types of tools will you need to migrate the calibration data?

Important topic but not much experience in this yet.

Need to migrate existing calibration data sets (parameters) onto the new platform. (Considering 13 RFCs the cost of adapting an exisiting simple NWSRFS operations will be small compared to the cost of re-parameterizing and re-calibrating thousands of segmentsl)

FEWS is NOT a calibration system; need to decide what will be the calibration system.

3. User Experience

3.1. What are the essential UIs that you need to maintain operability? Tell us about some custom UIs that you use. Which of these will need to be migrated before using FEWS?

TWXA (same idea as RiverPro): XWindow interface UI to generate text bulletins and graphical product for generating bulletins.

Snow Update: Mods to Snow Model on a monthly basis (RTI product used by NWRFC as well)

Time Series Display program WHFS suite.

Mountain Mapper Suite – used for QPF specification as well as precipitation and temperature data quality control.

QPF Comparison – Tool to visualize comparative data for QPF locations.

Temperature Plots – Interface to show spring temperature data and trends.

Historical Graphical RVF – Interface to generate previous forecast issuances along with observed values.

MM2RFS – Interface to collect, review, edit, and provide QPF and Rain-Snow Elevations to NWSRFS.

Mapper – Spatial display of time series data.

Rating Table Manager – Interface to ingest, manage, and distribute ratings.

Preprocessing

- *QC point precipitation and temps in geographic context*
- *Quantitative Precipitation Forecasts*
- *ESPADP (ensemble viewing -analysis and display program)*

3.2. How well do FEWS displays reflect information needed for the operational forecast process?

Limited experience. My sense is that additional displays will be required.

4. Extensibility and configuration

4.1. Tell us about the local apps that are essential to your operation. Which existing RFS and AWIPS visualization tools are being used as part of the FEWS pilot?

Tons of data preparation utilities constantly running to prepare data for preprocessor.

4.2. What is the complexity and stability of the process used to exchange data with other applications or databases (either local apps or partners)?

"Adequately complex"

4.3. Do you understand the level of complexity required to create data exports or data integration with other databases and/or applications? If so, is that level of complexity acceptable?

4.4. Do you understand the level of complexity involved in creating or integrating displays or visualization tools into FEWS? If so, is that level of complexity acceptable?

4.5. What concerns do you have about extensibility? How flexible is the pilot to adapt to process changes dictated by forecasting approach and philosophy at NWS?

5. Performance

5.1. What is the performance criteria and a reasonable time to run a forecast?

5.1.1. Perceived Application response time (rendering charts); is current RFS adequate?

Fair number of segments to go through. Current RFS performance adequate. RESSIM is quite slow. FEWS seems fine but depends on the model.

Data access and rendering information; ablility to provide customer on the phone with guidance. Important feature in RFS is that you can skip around very effectively to find specific points of interest (middle/end of forecast).

5.1.2. Perceived Backend processing speed(retrieving / storing data)

Ensemble processes need to be "reasonable" in terms of computational time.

5.1.3. Can the FEWS environment offer satisfactory response times when used by multiple (>3) forecasters simultaneously? Don't know.

5.1.4. What are the performance metrics/benchmarks that you would use to evaluate the new operational environment? We need some.

5.2. How does the performance of the pilot for conducting forecast runs compare to the existing forecast system, in terms of time and effort? This performance evaluation will cover both a stand-alone instance and a client-server instance of FEWS.

6. Training and support

6.1. How much training does new staff get before they begin to use NWSRFS?

New staff typically get 2 years training before solo-forecasting under important conditions. Lots of experience training new people. Pair forecasting for 6 months.

Training for calibration and operation models.

Concerns: FEWS is substantially more flexible (great) but there is a corresponding greater level of knowledge required to operate effectively.

6.2. How many people can you train at a time while maintaining operations?

Involve as many people in transition itself so they'll learn a bit during the migration.

6.3. What will be your approach for testing the system?

6.3.1. How many people involved in testing?

Will involve maximum possible staff in transition

6.3.2. Duration for testing?

Need follow on conversations to discuss scenario testing.

6.4. How long would you want the FEWS system to run in parallel before you feel confident?

6.4.1. Intensive and short process?

6.4.2. Slow gradual process?

High intensity and as brief as possible; depends somewhat on how they can bring parameter sets (and operations) into FEWS. The fewer the changes the higher the level of confidence.

Don't want to run dual operations any longer than absolutely necessary. (maybe bring RFS along in life-support mode as a backup).

6.5. Need training both in operations and in configuration?

6.6. What sort of support do you envision during migration?

Delft support has been terrific thus far.

6.7. How useable is the help system integrated with the pilot?

Online help vs. talking to people -> currently they talk to people.

Need an Implementation strategy related to building up a knowledge base across the RFCs.

7. Roles, Responsibilities, Staffing

7.1. Who will migrate your local apps? [OHD or you?]

7.1.1. Do you have any staffing concerns regarding development of local apps [java development]?

Dilemma is having full control of local apps to minimize coordination. Don't use the local apps database although local apps were registered for AWIPS II migration.

7.2. What general concerns do you have about this process?

7.3. What will be your team's interest in adoption of FEWS?

Very high and positive attitude.

7.4. Do you have concerns about OHD's role in FEWS?

7.5. Do you have any concerns about Delft's role

7.5.1. Anything Delft can/cannot provide?

7.6. Do you have any concerns about Apex's role?

Dialoguing with CAT team members develop this.

8. Features beyond NWSRFS

8.1. What other features or forecasting functionality beyond current state NWSRFS platform do you think will be cutting edge in the next 3 years that FEWS should be ready for (e.g. distributed modeling, ensemble forecasting)?

- *Ensemble forecasting is absolutely critical. FEWS implementation facilitates realization of XEFS.*
- *Handling the tides in the Sacramento-San Joaquin River delta and the mouths of several CA rivers..*
- *Improving the SS-SAC model.*
- *Water temperature modeling*
- *Hydro-dyanamic models (HEC-RAS)*

9. Deployment and System Management

9.1. What are your concerns about deployment related issues?

Even for the easy operations (like the arithmetic ones), it would be good to migrate to new platform so there can be a more straight-forward translation of the segments into the xml (or there needs to be significant help provided for this translation).

"Going to stretch us and we're ready."

Need more XML and Java; they're creative. Opportunity to do more efficient job if we look across RFCs.

9.2. What do you do today to support NWSFRS? How will it be different for FEWS?

9.3. What do you think about the suitability of the current hardware environment? We're very likely to need a significant jump in CPU cycles and storage.

9.4. How would client-server approach be deployed at NWS? Would you like the concept of a central server (any RFC would be able to forecast for any other RFC using central server data)? Possibly, but its quite unlikely that this will ever happen unless conditions force us. The individual practices and idiosyncracies of basins and water management practices require experience for proper operation. RFCs aren't likely to have the time to develop expertise in their neighbor's watersheds. The value would be greatest during a COOP where local staff would travel to another RFC to conduct operations.

9.5. Data Archival process

9.5.1. Describe your “Data archival process”

9.5.2. Do you need to access archived data? In what situation?

Not well orchestrated; need to find nexus between archiving and FEWS; really good attribute of FEWS is doing retrospective runs similar to WES.

9.6. How does the workflow management capability in the pilot match the current power of the hydrologic control language (HCL)?

9.7. Security issues:

Need to be reasonable and prudent related to security but total control doesn’t work. Need a reasonable set of security parameters.

9.8. Service backup

Improving ability to run off-site would be a good thing. COOP due at end of June (following southern region laptop model). Large network of local partners requires that CNRFC remain in the local area for collaboration..

10. Notes from Interview with Joe Intermill

Note that the Red Text are additional thoughts provided by Joe based on reading the notes taken by Apex.

1. End-to-end forecast process

1.1. Walk us through an end-to-end forecast process. What are some of the problems you encounter?

Come into the office; establish situational awareness on hydrology, meteorology, and integrity of computer systems (so there are many monitoring views for systems, resource utilization (i.e. disk space), batch process status. What’s happened over the last 24hrs, where are we now. Mostly use AWIPS applications although there are a couple home grown applications

Quality Control Stream Flow data QC: Make sure rivers are “behaving” (no flood situation). Precipitation data QC procedures (using Mountain Mapper software) both 24 hr and 6hr time steps. Precipitation data is large enough that they break up the data for multiple people (usually only up to four). Although for flood situations, they need to break up the data into lots of small chunks to QC. On quiet day it’s the opposite and 1 person can do the precipitation quality control.

Preprocessors run MAP- Mean Areal Precipitation, RRS - River Reservoir Snow moves data to operation data store.

Meteorologist (HAS people) on staff focuses on observed and forecasts temps (forecast freezing temp and forecast precipitation); 6 hourly time-step forecasts 10 days into future. Coordinate with WFO and HPC. ISC grids are used to share data. Preprocessors run for MAT (6 hour time step) and FMAP.

Now all the data is ready for model running. HAS person gives a briefing to the team (initial morning briefing) (5-15 minutes).

Anywhere from 2-7 people work on forecasts depending on the severity of conditions. Forecasters work through their IFP sessions and issue forecasts then monitor data to make sure no forecasts are very “out of whack”.

Collaborate with Corps of Engineers to combine natural forecast with reservoir outflow plans which are then incorporated into the overall forecast. COE is able to interact with modeling system (although they are only allowed to change reservoir outflows). They are able to see data either in plots or digital form.

Forecast output is generated in SHEF and distributed.

Many other products are generated from these data as well *(ex: QPF Plot, also 10-day plot).*

Flood outlook product: HAS discussion product (short, medium, long-term forecast).

Weekly products:

- *Suite of ESP products are generated every Monday/Tuesday.* Some products are distributed via ftp or the web.
- *Peak flow products for the various basins.*
- *AHPS products:* ESPADP is a critical program which performs statistical analysis on ESP time series.
- *STP: pseudo-deterministic forecast that goes out 45-60 days (loved by water resource managers)*

Best catalog of products is available on website.

Operational schedule

Weekday: In general there are 5 day shifts for hydrologists + 1 has person + 1 late shift. Weekends and Holidays will generally have fewer people (3-4) in operations and the number will vary seasonally.

Standard shift duty list. When flooding occurs, general areas responsibilities remain; but more people are brought into operations and forecast load is shared.

1.2. What issues would you foresee with the end-to-end forecast process implemented on FEWS given your experience with the pilot?

User mechanics need to be streamlined for ease of use.

2. Operation/Model integration

2.1. How identical would a new model have to be to the reference model to be acceptable (ex 90%)?

Models available in FEWS should possess (at a minimum) the same capabilities that currently exist in NWSRFS. FEWS currently has many modules that can perform similar functions required by our forecasting configurations. Some of these can easily be adopted without any re-calibration.

Training on the models is a big deal; so porting really complex models from a learning perspective is a good idea.

2.2. Walk through list of operations/models and for each describe the following:

See spreadsheet

2.2.1. Was it part of the pilot and if so did the model used during the pilot provide reasonable enough results that it could be used for real forecasts?

2.2.2. Is the operation/model required for your forecasting?

2.2.3. Do you know of an existing FEWS model that can be substituted?

2.2.4. How complex is this model? Is it possible to isolate this model or do we need to rewrite this model?

2.2.5. When this model is migrated, how many and which scientific deficiencies would you like to change for the model?

2.3. How do the models perform in manual vs. scheduled, background forecasting mode?

Models are almost exclusively run in an interactive mode. Recently, we have developed a batch mode for summer low flow situations. This method reduces forecast time, but still requires frequent interactive forecasting techniques when forecast review produces questionable results.

2.4. How critical is it for users to modify model states, model functionality and model operation during forecasting?

Absolutely critical to modify state. Very Critical!

2.5. How are the models calibrated in the pilot, and how much, if any, of the calibration data from the current forecasting system can be carried over into the pilot? What types of tools will you need to migrate the calibration data?

The NWRFC didn't perform any calibration for the pilot. We simply provided information about current basin configuration and historical calibration in the pilot area to RTi and DELFT. Specific questions on this process for the pilot should be directed to those groups. Questions about required calibration for widespread implementation are probably premature at this time. It is my impression that a major goal of this transition to FEWS is to minimize the requirement for that type of effort. SAC-SMA and SNOW-17 have already been ported. Assuming that common routing techniques (LAG/K, SARROUTE), and other major operations such as: DWOPER, FLDWAV, CONS_USE, etc. are ported as well, little or no calibration should be required. If any of these models are not directly ported to FEWS, significant calibration will be required.

3. User Experience

3.1. What are the essential UIs that you need to maintain operability?

Phase 1:

Something to interact with models: see simulation vs. forecast and update model states to generate a forecast.

Phase 2:

Other Displays for Precipitation & temperature & offline stream data, areal/basin average already exist so they could be used; see raw and Quality Control data.

Real-time monitoring of observations vs. forecast.

The browser-based system monitoring concept is very good. It could certainly be expanded to include other processes as well. *System monitoring (what he saw in pilot seemed outstanding).*

Calibration tools and procedures currently in NWSRFS are really quite extensive; not clear what is going to replace this. Our most recent discussions have centered around keeping the migration of NWSRFS calibration programs separate from the operational programs. In other words, they could have different migration schedules.

3.2. How well do FEWS displays reflect information needed for the operational forecast process?

Current displays need to be optimized for operations. This is not a "big deal". FEWS has adequate flexibility to configure the desired plots/displays.

3.3. Tell us about some custom UIs that you use. Which of these will need to be migrated before using FEWS?

Many of our custom utilities could possibly be replaced with current or "soon-to-be-developed" capabilities within FEWS. Most likely candidates would include data/forecast viewing utilities (FAVO, TS-Plot, DAT, DATGRAPH). The validation tool (XPLOT) could be replaced as well.

Our data quality control programs/utilities are candidates for a new look as well. Opportunities for replacement or enhancement with the FEWS implementation should definitely be discussed.

FAVO (forecast verification): look at a flow time series and go back and look at forecasts to see how you've been doing and then monitor observed data to see if it matches)

TS-Plot (standard AWIPS)

DAT – data viewer

DATGRAPH – data viewer

MountainMapper is used today for QC but moving to MPE *(AWIPS supported).*

TempQC – temperature quality control program

FMAP – utility which produces basin averaged forecast precipitation

CrossPlot (aka XPlot): regression of historical peaks for certain rivers regressed across basins; lets you use historical data from another basin to predict.

Utilities for updates to "Rating Curves/Tables" – gauge height vs. flow for a river.

System Monitoring was discussed earlier.

SETS – shef encoded time series program (produces shef encoded forecast products).

4. Extensibility and configuration

4.1. Tell us about the local apps that are essential to your operation. Which existing RFS and AWIPS visualization tools are being used as part of the FEWS pilot?

Local Apps were described in item #3. All of them can be used in conjunction with the pilot. Forecast from the pilot system are not currently being viewed. This will need to change in the very near term.

4.2. What is the complexity and stability of the process used to exchange data with other applications or databases (either local apps or partners)?

Not sure about this question. The processes are both complex and stable.

4.3. Do you understand the level of complexity required to create data exports or data integration with other databases and/or applications? If so, is that level of complexity acceptable?

The entire pilot configuration was done by RTI/Delft. Would want a bit more flexibility to add more data points more easily. It basically seems to work and be acceptable.

RTi seemed to struggle with multiple data sources for required data. The program probably needs to *be refined to provide more flexibility. In particular, allow multiple sources (SHEF) for input data.*

4.4. Do you understand the level of complexity involved in creating or integrating displays or visualization tools into FEWS? If so, is that level of complexity acceptable?

The exercises that we completed at the two day configuration training gave us a peek at what is involved. Like any process, it could probably be refined for clarity and simplicity. However, the current process is probably acceptable as a starting point. Effort is comparable to current OFS procedures.

4.5. What concerns do you have about extensibility? How flexible is the pilot to adapt to process changes dictated by forecasting approach and philosophy at NWS?

The utilities which use the OFS data files would have to change. The ones that hit the Postgres db may or may not need to change.

5. Performance

5.1. What is the performance criteria and a reasonable time to run a forecast? See 5.1.4. We have *Second/subsecond response time in current interactive mode.*

5.1.1. Perceived Application response time (rendering charts)

This needs to improve. It is not acceptable yet.

5.1.2. Perceived Backend processing speed(retrieving / storing data)

This seems to run fairly quickly, but obviously the current pilot is only a small part of our full-scale implementation. I suspect that it will need to speed up for the IOC.

5.1.3. Can the FEWS environment offer satisfactory response times when used by multiple (>3) forecasters simultaneously?

This has not been tested yet. However, this is absolutely required for all operational situations (especially floods).

5.1.4. What are the performance metrics/benchmarks that you would use to evaluate the new operational environment?

It needs to pass the "flood test". We would need to have multiple forecasters using the system in a "crunch-time" type scenario. Quantifying specs outside of that would be difficult. However, in that situation, forecasters could certainly tell you if it was fast enough or not.

5.2. How does the performance of the pilot for conducting forecast runs compare to the existing forecast system, in terms of time and effort? This performance evaluation will cover both a stand-alone instance and a client-server instance of FEWS.

Actual model runs were "okay", but a little slow. I am still *really learning* my *way around the mechanics so it seems slow.* I need to experiment more with making mods. At this point, I am still struggling somewhat with the mechanics. There seems to be a lot of flexibility. This is great, but this expanded capability also creates complexity which can slow folks down. This is something that we can refine with experience.

6. Training and support

I need to emphasize that most models are not changing. The forecast "system" is changing. Required training will still be extensive. However, if we were changing models (i.e. SNOW-17 and SAC-SMA) it could have been more extreme. National and local training plans will need to be developed. Similar to our current system (NWSRFS), we will need to have different levels of training (system level, configuration level, and operational level). Calibration training

will need to be provided in the future as well (we need a FEWS compatible calibration system first!).

6.1. How much training does new staff get before they begin to use NWSRFS?

Assume 0 experience with NWSRFS. Basin orientations etc is the same regardless of system. Series of tapes they go through (Eric Anderson), to orient them on the models (SNOW-17, SAC-SMA). They'll sit with an experienced forecaster and they'll watch for approximately 1 week. Then transition to working on their own with another forecaster right next to them (2-3 months). Then thrown into the rotation (pretty quick person would get into the full rotation in about a year).

Controlling factor is weather conditions; on a standard day it's very easy to train people up on the side (almost one to one trainer to trainee).

6.2. How many people can you train at a time while maintaining operations?

As long as it is a quiet time of year; we could have up to 3 or 4 trainees on the operational floor. *For NWRFC, they have a fairly regular annual cycle; summers (after snow has come off) July, August, and Sept are usually pretty quiet* operationally.

6.3. What will be your approach for testing the system?

6.3.1. How many people involved in testing?

6.3.2. Duration for testing?

Subgroup on the team will ensure integrity of system (output is sufficient).

6.4. How long would you want the FEWS system to run in parallel before you feel confident?

They went through this in the mid-90's once before in transition from SSARR to NWSRFS.

Once the new system was implemented for a certain area, one forecaster would be assigned to that area using the new system. Simultaneously, another forecaster would be working that same area in the old system as the primary (official) forecast. *Once they felt comfortable with the new system,* it would replace the old as primary.

The goal would be to get people comfortable with new system; convert to new system for operational forecasts and a few people stick on RFS to keep it up and running in case of catastrophic failure. Keeping NWSRFS up and running wouldn't be a big deal (1 or 2 people just making sure it had the data it needed etc.). It would be conceivable to keep RFS as a backup for an entire forecast cycle (likely a year)

.

6.4.1. Intensive and short process?

6.4.2. Slow gradual process?

6.5. What sort of support do you envision during migration?

Email is great but would really like phone support. It would be great to have the support person possess the capability to at the configuration with the developer..

6.6. How useable is the help system integrated with the pilot?

The help doesn't work yet; but hasn't been an issue because just referring to documentation has enough (docs have been good).

Effective communication methods will need to be employed between developers at all RFCs. This will allow us to learn from each other during the process (streamline implementation).

7. Roles, Responsibilities, Staffing

7.1. Who will migrate your local apps? [OHD or you?]

This is unknown. My guess is that certain local applications will be replaced by nationally supported software within FEWS. Other, more customized, software will likely be migrated by the individual RFC.

7.1.1. Do you have any staffing concerns regarding development of local apps [java development]?

While it would be good to have a common baseline of apps, it's likely that each RFC will want to customize certain applications.

Java programming is underway for many of the RFCs and WFOs. Expertise in Java is being developed as we speak.

7.2. What general concerns do you have about this process?

7.3. What will be your team's interest in adoption of FEWS?

Obviously, our interest is very high (our HIC is a member of the CAT team). We have a strong interest in helping the agency make a smooth transition to a new modeling system.

7.4. Do you have concerns about OHD's role in FEWS?

How much involvement will OHD have in this process? We hope that they will be very involved (especially with the porting of models, applications, etc.). We are also curious about OHD's ultimate role in support of the system (once it is implemented).

7.5. Do you have any concerns about Delft's role

None. I just hope that we can maximize their involvement. Their knowledge and experience with the system would make our transition MUCH smoother.

7.5.1. Anything Delft can/cannot provide?

7.6. Do you have any concerns about Apex's role?

*Who is going to be looking across the set of operations and deciding which ones will be converted or substituted, how will we prioritize the list of operations? (*answer: CAT Team)

We have a general question regarding support. Who's going to be the expert that we can call if FEWS is locked up at 1am? Different support levels and teams will need to be defined. This will obviously need to be answered during the transition.

8. Features beyond NWSRFS

8.1. What other features or forecasting functionality beyond current state NWSRFS platform do you think will be cutting edge in the next 3 years that FEWS should be ready for (e.g. distributed modeling, ensemble forecasting)?

- *Forecaster toolbox: something to give more confidence to forecaster during operations*
  - *Historic hydrographs from previous event and bring it into to current data to compare and see if historical trends should be considered.*
  - *Short recent history of forecasts shown next to current forecast to make sure trends are consistent*
  - Quick statistics on peak-to-peak and cross-basin relationships

- Quick access to observed/calibration/verification statistics

  - *Ensemble forecasting right there on the spot*
  - *Long term deterministic forecasts (beyond 21 days)*
  - *Data set visualization opportunities to enable significantly better monitoring tools.*
  - *Verification procedures and statistics (actual to forecast), identify biases at certain points.*
  - *Water resource program* – Results from ESP and regression techniques could possibly be produced and viewed together within FEWS*. Tie together flood and water resource forecasting in one place.*
  - **Get the archive database procedures improved and connect to FEWS system. Programs and utilities within FEWS could access and analyze this data (potential applications are numerous!).**
  - *Compare these priorities to Hydrological Program Priorities. Prioritize things that are common across all RFCs should go first; items that only apply to a subset should be reprioritized unless it is core to an RFCs operations.*

9. **Deployment and System Management**

9.1. What are your concerns about deployment related issues?

Training, Training, Training

9.2. What do you do today to support NWSRFS? How will it be different for FEWS?

OFS co-focal point…….probably similar role in FEWS (at first anyway)

9.3. What do you think about the suitability of the current hardware environment?

The hardware systems at the RFCs are currently inadequate and the FEWS implementation will be impacted by the performance issues currently affecting RFCs.

9.4. How would client-server approach be deployed at NWS? Would you like the concept of a central server (any RFC would be able to forecast for any other RFC using central server data)?

The C-S approach will be deployed at the NWRFC to emulate the same, multi-user computing environment that exists today. This C-S can be expanded in the FEWS-era to consider a multi-user and a distributed computer environment that will employ backup capabilities. The concept of a central server certainly has merit but the current approach of future AWIPS will not lend itself to a properly configured, performance-based system due to constraints on hardware and broadband communications.

9.5. Data Archival process

9.5.1 Describe your “Data archival process”

9.5.2 Do you need to access archived data? In what situation?

Described and addressed in other sections within the survey

9.6. How does the workflow management capability in the pilot match the current power of the hydrologic control language (HCL)?

At first glance, it appears to be fine. Further implementation experience will "tell the tale".

10. Final Thoughts

Pilot next steps:

- *Need to take steps to ensure all Pilots are configured as they would be in the IOC; this should be done ASAP. (Reservoir modeling system is currently insufficient).*
- *Need ability to produce a forecast out of FEWS that is exactly the same as what's generated out of* NWSRFS *so easy direct comparisons can be made.*
- *Better criteria for gauging success of the pilot sites (performance etc).*
- *Test true operator client mode.*
- *Need some training for how to most efficiently use the operator GUI;*
- *Operator UI Gap Analysis: do a side by side comparison of current IFP and FEWS and do a quick GUI gap analysis.*
- *Additional round of implementation training – 2.5 days of FEWS training is not* sufficient. *We need another round that incorporates training on (*not yet developed*) tools that will streamline implementation; (Example: easy tools to transfer data, parameters, and configurations from OFS* and load them *into FEWS).*
- *Conversion tool gap analysis (see previous bullet).*
- *Must get pilots fully up and running; creates momentum and buy-in. Confidence momentum.*
- *Need to decide on which XML Editor is going to be the standard*
- *Which environment is the right environment for development (all development is done in Linux environment)?*
- *Currently all the pre-processor work* (MAP,MAT,FMAP,FMAT) is done outside of FEWS (this includes estimation)*. Is this the future vision?* Should we port/develop pre-processors within FEWS*? (Currently part of NWSRFS).*
- Should we migrate our quality control programs to FEWS*? (currently not part of NWSRFS)?*
- We need to finalize and implement a plan to either migrate DWOPER and FLDWAV to FEWS or convert to a new dynamic model and begin recalibration efforts. Significant resources will be required for either option to be successful.
- A decision will need to be made concerning the porting/replacement of ESP and ESPADP capabilities.

11. Notes from Interview with Sudha Rangan

Once things are deployed, end users should not be able to touch some configurations, but Sudha and group should be able to touch those parts. Debug mode. So only model developers will be able to change diagnostics not end users. Karel: For migration and support. May need more specific determination of which parts are off limits for end users.

Define the boundary between system configurations (to be delivered during deployment) vs. user modifiable configurations.

Standalone versions will be easier than embedded ones. Almost all the models we will be converting will be embedded ones, but once we document the process of extracting/converting models from NWSRFS, it should be fairly smooth sailing.

1. Extensibility and configuration

Parameter conversion. Need well defined process. Need a tool that will automatically convert parameters for each model and possibly convert workflows as well given the large number of basins.

2. Training and support

While developing plug-ins for more MODS, documentation is not robust. Need more of that; No experience developing plug-ins; need to start building experience here. Not much information on how to build a screen in to the FEWS GUI.

Problems/Feature requests have been casually mentioned; kept track personally; communication via email. Karel: They have a working procedure for maintaining these. They will need to open up the development environment – think of a procedure for that. (This basically was calling for a shared development environment – i.e. one that Delft would be able to get on to – ultimately, to be able to help with the process of migration and possibly working with FEWS code as well – in the absence of this, Delft folks may have to be on-site to help).

Need online chat tool (with Delft) for debugging. Karel: maybe positioning someone in NOAA offices if that works better. Some dedicated time for support.

3. Deployment and System Management

Adapting models is fairly straight forward. The greater challenge is extracting the model from NWSRFS – though this is the greater challenge, it is more a question of having a well-defined and documented process for conversion including some documentation that people can refer to with respect to testing the model in its old environment to compare against the newly converted model. Once some effort has been spent on this, it will be much quicker for everyone who works on model conversion to be productive. FEWS configuration (XML)?

Having conditional statements allows it to be powerful – allows it to control things, combination of things, diff things in diff scenarios maybe impossible to control. Karel – set some conditions but there is a limit. This was just an observation about the fact that there a couple of key features in FEWS that it would be useful to have at a future date.

Process of adapting to FEWS is the simplest. Simplest schema. This is not a problem. Not hard to adapt to FEWS was the comment.

There is an issue with ensembles aware/unaware mode published interface PI schema. We currently have to use a workaround to transmit ensemble id information from FEWS to an external module currently and are hoping that in the future Delft will correct this by enhancing their Published Interface.

12. Notes from Interview with Pedro Restrepo

1. **Extensibility and configuration**

Parameter conversion - Need well defined process.

2. **Training and support**

Issues:

Assumption: people developing models in FORTRAN will be able to continue to do so (is this correct?).

Who is going to be doing the adapter; what sort of training is required to create the adapter?

Who is going to provide technical support for Science branch so they don't have to be FEWS experts? Not sure how much training/technical expertise is required to get functional in FEWS.

3. **Roles, Responsibilities, Staffing**

Research is clearly not as critical because there is not a daily operational output but at the same time don't want to have research grind to a halt because FEWS is in the field and not used by research team.

One good thing would be to have Delft visit OHD and talk to science guys to get input on the transition.

It is clear that some RFC's have unique problems (example Alaska). But is it necessary for these applications to be outside the FEWS system. Institutional problem of how developing RFC's can notify other organizations know about it and use it. Today, this happens at bi-annual/regional meetings.

Is there a corollary to "local applications" in the FEWS environment?

There needs to be a way to "tag" each application as either "local" or "official".

What would be the process of a new model being distributed to a select set of RFC's who could "test" it in non-operational environment, make enhancements, publish enhancements back to OHD to incorporate and publish a new official model. How can CHPS/FEWS streamline this process?

4. **Features beyond NWSRFS**

Ensemble models and supporting this (not certain whether, how well this is currently supported)

Multi-model ensemble models: XCFS project – ensemble preprocessing (accept all inputs and produce inputs for xefs2 ensemble model); ensemble post-processing; ensemble verification.

Want to generate calibration data from same system as generating forecast to streamline/unify the calibration data. Example, currently if a forecaster generates a forecast from a calibrated model and verifies outputs and finds that it is not good; they need to re-calibrate the model and do a "hindcast" then verify that improved calibration results in improved results.

Data simulation (they know Delft is doing something on this but not sure if this is sufficient, the guess is that it is only a small subset)

Operations currently are one step at a time

Add time step in the past to the operation – FEWS must support ability to look back in time beyond 1 time step. Two methods of linking models mechanism:

1) *Get all data for model and run model with all time steps then the environment takes this data and runs next steps (another model or post-processing).*

2) *Other option is for the each model to be run for each time step. => Complexity is that the model has to look back into time.*

3) *It seems like FEWS works with option 1; there may be the need for option 2 but not a concrete example of this yet.*

Distributed model:

How does Calibration play with the Distributed model; calibration is not part of FEWS but it is critical for the success of distributed model. Set parameters; run model; compare output to measured reality; updated parameters.

Comparing iterations/updates to models: compare outputs of a new version of a model to a previous version of model.

Ensemble product generator – create ensemble products

To be able to experiment with new "models"/other technologies and provide verification of data: need to be able to create not just "forecasts" but also "hindcasts".

Need to allow researches to make changes to "experimental" models without impacting "production" models.

Data simulation could be a replacement for Runtime modifications: change model state variables to observed states. However, data simulation may not be a complete replacement; still need to provide the ability for human forecaster to add additional modifications based on experiential data.

Researchers (both internal and external) need to be able make modification and do experiments without impacting production operational models. Inputs may be the same but the outputs need to be sent to a different place and ensure that operations do not use these outputs.

Big sales point in FEWS is to make it easy for operations to move to a newer version of the model as quickly/easily as possible.

Who is responsible for system level code: Enhancements to FEWS to support "Scientific-Research"; Need a protocol of submitting these enhancements and who is responsible / how it will be done.

13. Initial Inventory of Operations, Preprocessors, and Local Apps

13.1 Pre-processors:

1. Mean Areal Precipitation (MAP)
2. Future Mean Areal Precipitation (FMAP)
3. NEXRAD (Gridded)
4. Mean Areal Precipitation (MAPX)
5. Mean Areal Temperature (MAT)
6. Future Mean Areal Temperature (FMAT)
7. Mean Areal Potential Evapotranspiration (MAPE)
8. River, Reservoir and Snow (RRS)
9. Preliminary Precipitation Procesing Program (PXPP)
10. National Digital Forecast Database (NDFD) to Future Mean Areal Temperature (FMAT) Preporcessor (NDFD2RFS)

Not every office uses all of these programs and some may not be regularly used at all in operations.

13.2 Operations

This table describes the set of operations in use along with a listing of how difficult it would be to migrate off of a given operation and what the current status of the operation is in the overall FEWS migration.

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
ADD/SUB	Basic Arithmetic					In FEWS. but no user interfaces to specify these things	Medium (user interface changes are needed)
ADJUST-H	Time series					In FEWS. Can be achieved via transformations/arithmeticFunctions etc.	Medium (if needed in the same way)
ADJUST-Q	Time series		Not complex		Low	In FEWS (partially) *(same as above)*	Low
ADJUST-T	Time series (Tide adjustment)		Complex		Medium	Not in FEWS	Low (changes to tide time series based on user input)
API-CONT	Model				will compare with FEWS	Not in FEWS	Medium to High (because of mods)
API-HFD							Medium to High (because of mods)

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
API-MKC				Don't actually use this operationally			Medium to High (because of mods)
BASEFLOW	Allows modification of baseflow			Critical			Low-Medium (because of mods)
CHANGE-T	Time Series Utility		Not complex		Low	In FEWS	Low
CHANLOSS	Model Channel Loss	Required (cannot conceive of moving to a different model available in FEWS since its so straightfor ward)	Not complex		Low	In FEWS	Low
CLEAR-TS	Time Series Initialization					In FEWS	Low
CONS_USE	Model -Irrigation driven mechanism for projecting losses/gains in a basin (better version of CHANLOSS)	Required					Low
DELTA-TS	Time Series Utility	Required					Low

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
DWOPER	dynamic wave op	Lots of recalibration and training to move to a different wave crest model		Precursor to FloodWave (primarily use this on lower part of MS river, problems converting to FloodWave that never were resolved), whether or not this is needed is dependent on whether FloodWave was supported and whether they can convert to HEC-RAS			High
FFG	Flash Flood Guidance		Erroneous	There is work going on to go a diff route with FFG, in process testing RDAHM to replace FFG (not sure if this will be successful)	They moved to an external utilty; this needs re-examination		High (whole application around it - need to make sure everything cared for)
FLDWAV	Flash Flood Wave	Some stability and other historical problems have prevented moving across the board		Primary model for tracking flash flood cresting, replace DWOPER where possible			Very High (whole application around it - need to make sure everything cared for)
GLACIER							Low

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
LAG/K	Routing utility	Used in calibration process; so changing to different routing model would require re-examination of calibration data	Not complex	Shouldn't be a big deal, but pretty popular method for doing routing. Wide use means it would require lots of work to migrate; although excited that FEWS brings more routing schemes	Low	Not in FEWS	Medium
LAY-COEF	Another routing (layered) technique			Not sure how widely it is used			Low
LIST-FTW						In FEWS. Need to see if it has some sort of print facility	Low (if necessary)
LOOKUP	Used to to tabular modeling	Required	Not complex	Critical	Low	In FEWS (should be able to do this with FEWS transformations/functions)	Low

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
LOOKUP3	Used to to tabular modeling	Required	Not complex	Critical	Low	In FEWS(should be able to do this with FEWS transformations/functions)	Low
MEAN-Q	Time Series Utility (Discharges to period avg)	Required		Critical	Low	In FEWS (with transformations etc.)	Low (should be able to use what FEWS has)
MERGE-TS	Time Series Utility			Critical		In FEWS	Low to Medium (if MOD GUI needed)
MULT/DIV	Basic Arithmetic					In FEWS	Low
MUSKROUT	Routing Technque (Muskegam technique)			Need to figure out how comparable it is to FEWS muskegam routing model; the existing NWSRFS one is not very complex		In FEWS	Low
NOMSNG	Time Series Utility (NoMissing)		Not complex	Critical	Low	Not sure	Low
PLOT-TS	Visualization - Used for Callibration	Assuming it would be replaced by FEWS		Not sure this is really necessary, replaced by other FEWS functionality	Likely to be idiosyncranic to NWSRFS	In FEWS (Looks like covered)	Medium (If any changes need to be made to FEWS GUI)

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
PLOT-TUL	Visualization	Assuming it would be replaced by FEWS		Not sure this is really necessary, replaced by other FEWS functionality	Likely to be available in standard Fews; probably not required	In FEWS	
RES-J	Model (multi-reservoir method that handles dependencies)		Very complex	Pretty Important; performance in operational/daily mode is not too bad. Very problematic in EFP mode (particularly bad because of single threaded nature of NWSRFS), but it definitely helps with the forecast	Likely to be available in standard Fews; probably not required		Very High (also, ResJ is in C++ - a lot of C code as well, need to see what this is about)
RES-SNGL	Model (replace missing time series pts)		Complex	Pretty Important	High		High
RSNWELEV	Produces Rain/Snow elevation for use in model	Absolute requirement to have something similar, but moving to an equivalent wouldn't impact	Not complex		Low	In FEWS	Low

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
		calibration/training wouldn't be too complex					
SAC-HT	Model (replace SAC-SMA)			In process of evaluating whether it can replace SAC-SMA		In FEWS (HT also available but needs more work)	Low (only estimate for additional work needed to bring Sac-HT/SacSma to operational level - SACSMA-HT should actually be a separate operation)
SAC-SMA	Model	Already ported to FEWS		Likely to be replaced to SAC-HT, although not 100% sure it will 100% replace	Done	Not in FEWS	Medium
SARROUTE	Routing utility (ported from SAR world)	Used in calibration process; so changing to different routing model would require re-examination of calibration data				In FEWS	Low (should be able to use as is in FEWS)
SET-TS	Time Series Utility				Low	In FEWS	Medium (needs more work to include mods)

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
SNOW-17	Model	Already brought over		Important but althready there	Done	Not in FEWS	High
SSARRESV	Reservoir Model	Not having this would be an issue, conversion to another model would be a big deal; better to port.					Medium
SS-SAC	Model, Needs another round of refinement; DJ's data assimilation work		Very complex		High	In FEWS (partially)	Medium
STAGE-Q	Time Series Utility (uses rating curve)				Low		Medium
STAGEREV	Post processing of forecast Review differences (handles tidal impacts) between simulations and observed; part of process with DWOPER/FLDWV	Critical					Low
TATUM	Routing Technique			Most widely used for 784 operations			Low
TIDEREV	Review differences between projected and observed tides; part of process with DWOPER/FLDWV	Critical	Not complex			In FEWS (partially)	Medium (but towards low side)
UNIT-HG	Converts precip into flow into river	Critical		Important	Low	In FEWS	Low (should be able to use as is from FEWS)

Model	Description	NWRFC	ABRFC	NCRFC	CNRFC	Current Status	LOE (based on code)
WEIGH-TS	Time Series Utility, used to unequally combine contributions of sub-areas of watershed				Low	In FEWS	

13.3 Local Applications

This describes the full set of Local Applications collected with the Local Applications that are reported to being actively used by the NCRFC, ABRFC, or NWRFC listed first and highlighted.

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	RHA	LDM and associated processes	critical	used mostly to share meteorological data between research communities. Dedicated sharing mechanism allows people to subscribe to specific products. Allows digital sharing of time series data.	Y		
CR	MSR	Ftmp	critical	Forecast Temperature Modification Program. Deals with temperature data quality control but hopeful that within next several months (by next winter), transition to GFE (gridded field editor) for this.	Y		
ER	RHA	NMAP and supporting utilities	critical	Forecast rainfall. QPF creator; NMAP provided by NCEP; other code provided by other RFCs and MARFC.	Y	Y	
SR	TU	xnav	critical	data visualization tool	Y	Y	

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
SR	TUA	xdat	critical	data viewer/editor. table perspective of db tables GUI for interacting/viewing/editing data within the local db (IHFSDB).	Y	Y	Y
ER	RHA	XSets/ISets	critical	RVF preparation utilities; XSets is MARFC version; ISets is graphical editor		Y	Y
CR	MSR	OPIE	critical	Operational Integrated Environment - Forecast Editor. used for generating forecasts; visualization tool for creating forecast product. Expectation is that this would be replaced within FEWS.	Y		
		D2D		Baseline AWIPS application which visualizes data for situational awareness of current conditions.		Y	
		(P3)		instead of baseline AWIPS app (MPE) for radar precipitation.		Y	
		GIS tool		(for development and operations such as FOP)		Y	
WR	RSA	ifpcom	critical	interactive forecast program			Y
		fcst_prog		Mmonitor status of river forecasts		Y	Y
SR	ALR	shiftlog	critical	Logs shift information		Y	Y
WR	PTR	favo	critical	Displays observed hydrologic data along with up to days of forecasts (look at a flow time series and go back and look at forecasts to see how you've been doing and then monitor observed data to see if it matches)			Y
		DATGRAPH		data viewer			Y
		MountainMapper		used today for QC but moving to MPE (AWIPS supported).			Y
WR	PTR	tempqc	important	temperature qc program			Y
WR	PTR	fmap	critical	converts point QPF to Areal QPF (utility which produces basin averaged forecast precipitation)			Y

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	PTR	xplot_prog	critical	cross basin peak stage comparison program			Y
CR	KRF	AASets	critical	reads NWSRFS binary files and creates data file			
CR	KRF	afos text	important				
CR	KRF	AHPSPRODLIST	critical	Manage AHPS forecast points			
CR	KRF	arcfcstprog	important	displays obsd data and past fcsts from achive db			
CR	KRF	archive programs	critical	move various data into proper place on archive server			
CR	KRF	arcID.f	Nice To Have				
CR	KRF	batch_ofs	critical	user interface for batch ofs			
CR	KRF	biasdata.pl	important	Create Web page of radar bias information			
CR	KRF	bias_reset.py	important	Graphical display and edit of Bias and ZR data including hourly gage values			
CR	KRF	bin	important	Data management software for AHPS and calibration			
CR	KRF	calb_logs	important	Logs parameters tried and resulting calibration stats			
CR	KRF	calcfzdd	Nice To Have	creates accum freezing degree day plots			
CR	KRF	calcpc	important	calculates PPD from PCI data			
CR	KRF	callibration utilities	Nice To Have	collect data from archive server and move into appropriate calibration directory			
CR	KRF	canal.f	Nice To Have	converts Nebraska canal data to OH card format			
CR	KR	check.hads	critical	Checks HADS products for new files and warns			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	F						
CR	KRF	check.pl	critical				
CR	KRF	cocoras.pl	important				
CR	KRF	Collect.pl	critical				
CR	KRF	convert_basins.pl	critical	convert Basin Boundaries from OFS to various other formats			
CR	KRF	Corps Data transfer	critical				
CR	KRF	create_crest	important	creates SHEF data message for NCDC			
CR	KRF	crest_verif	important	Does stats for crest forecast verification			
CR	KRF	daily_msg.pl	critical				
CR	KRF	data4maps	critical	scripts that create input files for gis and other apps			
CR	KRF	data4web	critical	creates obsd* fcst status files for use by gis			
CR	KRF	DataAnalysis	important	Data management software for AHPS and calibration			
CR	KRF	datplot	Nice To Have	plots observed data			
CR	KRF	dbadmin	Nice To Have	mis DBA scripts			
CR	KRF	dbpg.pl adpg.pl	important				
CR	KRF	dcp_pcpn/	Nice To Have	apps that generate info about pcpn rpts			
CR	KRF	defrat	critical	Define to rating in OFS			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	KRF	defsta	critical	Define new station in OFS			
CR	KRF	discharge-data	Nice To Have	Create historical time series and freq curves			
CR	KRF	droughtmonitor	important	creates input file for gis			
CR	KRF	ESPprecipcompare.f	Nice To Have	Shows edited differences between two OH card sets			
CR	KRF	ESPutilities	critical	Create imput files for AHPS probability forecasts			
CR	KRF	fcstptlist	critical	Create lists of forecast points			
CR	KRF	ffg_interface	critical	interface to select FFG options			
CR	KRF	fieldgen.pl	Nice To Have	GUI for re-running Fieldgen for selected periods			
CR	KRF	FIOP	critical	Flop.tcl			
CR	KRF	fixtime.f	Nice To Have				
CR	KRF	fixtime.f/hsa.f	Nice To Have				
CR	KRF	ftemps	critical	create SHEF files of future temperatures from CCF files			
CR	KRF	gage_control.py	critical				
CR	KRF	get_stlrrmmci.pl	critical	get RRM message from EAX via/ldad. Transmit			
CR	KRF	get_zr	important	create web page of radar ZR information			
CR	KRF	hcm.pl	important	GUI for creating and transmitting HCM messages			
CR	KRF	herm_xfer	critical	Create RRM product of Hermann flows for NCRFC			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	KRF	hotspots	important				
CR	KRF	juldat*.f	Nice To Have				
CR	KRF	local cgi web pages	critical	9 perl programs to create interactive web pages for display in AWIPS			
CR	KRF	Local Web Pages	critical	7 perl programs to create Status web pages for display in AWIPS			
CR	KRF	manual_pst	Nice To Have	Prompts user for dates and posts data to OFS			
CR	KRF	maxmin	critical	Create SHEF file of max/min temperatures from hourly database values			
CR	KRF	mbrfclib	critical	fortran subroutines and c functions used			
CR	KRF	meltadv	critical	interface to setup input files for spring outlook			
CR	KRF	misc_support	important	misc support apps			
CR	KRF	modclean	critical	Clean out old mods from OFS mods files			
CR	KRF	Mover.pl	Nice To Have	Exchange datas from PC network through ldad			
CR	KRF	mpe24.py	Nice To Have	Perl and C programs for editing 24-hour MPE fields			
CR	KRF	msg4ncdc	important	creates SHEF data message for NCDC			
CR	KRF	ofs_xref	critical				
CR	KRF	other	important	Misc OFS support programs			
CR	KRF	outlook_ofs	critical	Interface to run Spring Snowmelt Outlook			
CR	KR	package	critical	Interface for creating the AHPS probability			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	F			forcasts			
CR	KRF	parse_rr7	critical	parse rr7 collectives and extract needed hourly data			
CR	KRF	pcdly	critical	Creates list of daily precip values from IHFS			
CR	KRF	pcmag	critical	Creates list of daily precip values from OFS			
CR	KRF	pcpn_q_c	critical	QC precip to be posted to OFS			
CR	KRF	pcpnrpt	Nice To Have	generates pcpn report			
CR	KRF	Pcrain.f	Nice To Have				
CR	KRF	potshots	critical	creates input data files for flop apps			
CR	KRF	printops	critical	Retrieves SACSMA and SNOW states from OFS			
CR	KRF	psycopg	important	open source python library for Postgres			
CR	KRF	qpf programs	critical	Exchange datas fromPC network through ldad			
CR	KRF	ratable	critical	Creates files from OFS ratings			
CR	KRF	rate_shift	critical	Shifts OFS rating and creates log of changes			
CR	KRF	raw_support (xset_allret)	Nice T oHave				
CR	KRF	read_mapx	critical	Runs hourly mapx and checks output			
CR	KRF	read_ofs_ts	important	Retrieves SACSMA and SNOW states from OFS			
CR	KRF	regionalparms.f	Nice To Have				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	KRF	resendQPF.pl	Nice To Have	allow retransmit of QPF data on WAN using GUI			
CR	KRF	reservoir	critical	7 apps that SHEF encodes USACE and USBR data			
CR	KRF	rsync_mpe.sh	critical	script for rsyncing MPE data for national precip gage			
CR	KRF	RUHT Notify	critical	rating * unit hydrograph transfer apps used by most WFOs east of the Rockies			
CR	KRF	RUHT Notify	critical	rating and unit hydrograph transfer apps (rfcside)			
CR	KRF	Run_allfgs_no_qpf_LX	critical	Run all forecast groups on no_qpf OFS files			
CR	KRF	run_allfgs_no_qpf_nonrout ...	critical	Run all forecast groups on no_qpf OFS files			
CR	KRF	run_final	critical				
CR	KRF	Runshef.pl	important	GUI for creating SHEF products			
CR	KRF	rvrassign	critical	Creates river assignments report			
CR	KRF	rvr_verif	Nice To Have	local verification apps			
CR	KRF	sasm	important	Creates summary of synoptic MET stations			
CR	KRF	scripts	important	misc support scripts			
CR	KRF	seg2mcp3	critical	Converts OFS segment definitions to calibration decks			
CR	KRF	send_rfc_products	critical	nations program for transmitting RFC products on WAN			
CR	KRF	send scripts	critical	send gridded data to national centers on WAN			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	KRF	shef_chek	critical	apps that sorts decoder msg output files by parsing and posting errors			
CR	KRF	snowdepth	Nice To Have	Creates summaries of snow data			
CR	KRF	snowtel	critical	12 apps that generate SHEF encodes USACE and USBR data			
CR	KRF	ssm.pl	critical	parse Canadian SSM collectives			
CR	KRF	stations	critical	creates input file for other apps			
CR	KRF	strange	critical	Finds station to post strange precip values			
CR	KRF	strangers_list	important	creates list of stranger station pcpn rpts			
CR	KRF	switch_map	Nice To Have	Creates mods to switch from mapx to map			
CR	KRF	sycn.pl	critical	sync images to ldad for web page			
CR	KRF	task and run info	important				
CR	KRF	toExcel.f	Nice To Have	alligns values from several OH card sets for Excel plot			
CR	KRF	to_wan	critical	Configures products to send to the wan			
CR	KRF	TSvect2.f	important	Puts OH datacards into readable format with day show			
CR	KRF	USBR Data Tranfer	critical	4 scripts to transfer data to the USBR via/ldad			
CR	KRF	usgs2ofs.f	important				
CR	KRF	USGSmissing.f	important	creates OH crads of estimated USGS streamflow only			
CR	KR	usgsmonth.f	important	USGS mean daily flow to monthly totals for			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	F			fastectc data			
CR	KRF	utility and research scripts	Nice To Have	various perl programs for utility research			
CR	KRF	verify	important	misc support scripts for river verification			
CR	KRF	volumeAF.f	Nice To Have	USGS mean daily flow to monthly AF			
CR	KRF	watersupply	critical	water supply precip table and data for maps			
CR	KRF	worksched	critical	Creates work schedule			
CR	KRF	wspinpdb	critical	retrieves PPM data from watersupply db			
CR	KRF	xbatch.pl	important	GUI for running batch ofs and viewing log files			
CR	KRF	xgensum	critical	adds plain english summary to awips baseline XSETS generated fcst messages			
CR	KRF	xnav util program	important	create XNAV overlays			
CR	MSR	archive_monitor	Nice To Have				
CR	MSR	calb_sta_data	Nice To Have				
CR	MSR	canmet	important				
CR	MSR	canrad	important				
CR	MSR	checker	critical	System status monitor			
CR	MSR	checkstg	important	Stage Checker			
CR	MSR	clicker	critical	Gage Removal Tool			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	MSR	convert_vgf	critical	Data file converter script			
CR	MSR	cpc	important	Script to move CPC images from dir to dir			
CR	MSR	dambreak-dam	Nice To Have	Coversion Tool for dambreak utilities			
CR	MSR	data4maps	Nice To Have				
CR	MSR	degree	Nice To Have	Conversion utility for mapping			
CR	MSR	doahps	critical	Front end for AHPS processing			
CR	MSR	esg	Nice To Have	Front end for ESG processing			
CR	MSR	fldwav	important	Flood wave prep scripts			
CR	MSR	getWebData	important	Prep scripts for obtaining DNR data from web			
CR	MSR	hcm	Nice To Have	Generates HCM message content			
CR	MSR	hmd	Nice To Have	Generates HMD message content and manager			
CR	MSR	iac_gui	Nice To Have	Interagency Coordination GUI			
CR	MSR	ice_jam	Nice To Have	Icejam Utility software			
CR	MSR	ifp_comp2	important	Modified IFP Companion			
CR	MSR	mapPlot	important	MAP Plot Utility			
CR	MSR	mapxPlot	important	MAPX Plot Utility			
CR	MSR	mastercheck	important	MPE Check Program			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
CR	MSR	mat_plot	critical	MAT Plotting program			
CR	MSR	mis_check	critical	MIS forecast auto checker			
CR	MSR	ndfd	critical	ND Flood mapping Scripts			
CR	MSR	obsget	critical	NCRFC Custom OFS data extraction scripts			
CR	MSR	ofsmenu - customized	critical	Access OFS functions			
CR	MSR	oper	important	Oper Log Menu System			
CR	MSR	QPFC	critical	Ensemble Forecast Generator			
CR	MSR	QPFC_GRAPHICS	critical	Ensemble Forecast Generator - AWIPS Baseline RLS			
CR	MSR	Rcurve	critical	Rating Curve Management			
CR	MSR	River_monitor	important	River Status Monitor			
CR	MSR	RUHT Notify	critical	Rating curve transfer program to WFO			
CR	MSR	rvfmonitor	Nice To Have	River forecast monitor scripts.			
CR	MSR	staffing	important	Staffing decision tree GUI			
CR	MSR	v5oper	critical	Forecasting Tool			
CR	MSR	webmaps	critical	RFC Mapping software for public website			
ER	RHA	Alarm System	critical	Critical message handling; anything operationally significant			
ER	RH	Data Flow Management	critical	Monitor and control RFC data flow			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
ER	RHA	Flood Outlook Product	critical				
ER	RHA	HAS Data Handling and Qua ...	critical	View and edit operationally critical precip and temperature data			
ER	RHA	Map Manager	critical				
ER	RHA	Miscellaneous AWIPS Produ ...	critical				
ER	RHA	Miscellaneous NWSRFS OFS ...	critical	IFP Utilities; OFS Output Viewers; Data Input Utilities			
ER	RHA	OFS Management	critical	Access to NWSRFS OFS Utilities			
ER	RHA	OFS Mods Management	critical	Manage NWSRFS OFS Run-time MODs			
ER	RHA	Queuing System	critical	Daemon providing queued execution of submitted jobs on multiple boxes			
ER	RHA	Station Log	critical	Log of Operational RFC Activities			
ER	RHA	Transmit Manager	critical				
ER	RHA	Various cron processes	critical				
ER	RHA	Various data handling dae ...	critical				
ER	TAR	Adirondacks graphics	important				
ER	TAR	CanadaMPE	critical	Posts Canadian Radar data for use in MPE			
ER	TAR	Checklist additional GUIs	important	3 GUIs accessed from Shift Checklists			
ER	TAR	Checklist Summaries (5)	important	Generates summary of shift duties checklist			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	TAR	Checklist supporting scripts	important	Scripts to support Checklists (8)			
ER	TAR	Contingency	critical	GUI to aid forecasters in generation of contingency forecasts			
ER	TAR	Dambreak Rules of Thumb	Nice To Have	Rules of thumb for forecasters in case of dambreak emergency			
ER	TAR	DataMonitor	important	monitor flow of necessary products into RFC and for processing by NWSRFS			
ER	TAR	DataQC	important	GUI to assist in the display and QC of station obs			
ER	TAR	GrassMap	critical	Generates web maps of point data extrapolated over space			
ER	TAR	GRASS Watersupply scripts	critical	2 Programs to prepare watersupply products with GRASS software			
ER	TAR	hcm	critical	GUI to assist forecaster in generation of HCM product			
ER	TAR	HLT scripts	critical	Cron scripts that update data for the Hurricane Liaison Team			
ER	TAR	HMD	critical	interface to aid forecaster in generating HMD product			
ER	TAR	HYDgen	critical	cron script to generate HYDTAR product			
ER	TAR	hydsum	Nice To Have	sums up several days of HYD product to get storm total			
ER	TAR	IFP Companion	important	gui to provide additional support info while running nwsrfs ifp			
ER	TAR	Logfile manager	Nice To Have	Assists forecasters in reviewing a variety of data processing log files			
ER	TAR	MPE Bias Reset	critical	Resets MPE bias values to 1.0			
ER	TAR	MPE scripts	critical	Process MPE QPE data and transmit gridded data to users			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	TAR	NERFC Alarms	important	Monitors system and data for alert conditions for forecasters			
ER	TAR	NMAP graphics - Grass	important	Generates QPF graphics fron NMAP with GRASS Software			
ER	TAR	NOS scripts	critical	Provide data feed to NOAA/NOS for observations/forecasts for New York Harbor			
ER	TAR	NWSRFS Scripts	critical	A collection of scripts to aid forecaster in making regular NWSRFS runs			
ER	TAR	pc2pp	Nice To Have	program to convert PC to PP data for use in MPE and other data qc procedures			
ER	TAR	QPF verify	Nice To Have	GUI to display local QPF verification graphics			
ER	TAR	ReadOFS	critical	Read and display time series from NWSRFS data base			
ER	TAR	RFC Backup scripts	critical	Collection of scripts to keep RFC backup data and systems current			
ER	TAR	River Conditions	critical	Creates Web displays of current and forecast river conditions			
ER	TAR	RiverWatch	Nice To Have	GUI to monitor river levels near critcal stages			
ER	TAR	rvfplot	important	Reads the RVF and generates a gif image			
ER	TAR	SHEF decoder	important	Standalone SHEF decoder to create text data file rather than posting to RDBMS			
ER	TAR	Shift Checklists (5)	important	Shift duties checklists			
ER	TAR	SLOSH	important	Provides display of tidal forecasts in advance of Hurricane			
ER	TAR	Station Log	important	Provides interface to office station logs			
ER	TAR	Watersupply Perl programs	critical	4 Programs to prepare and send ESS and ESP products			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	TAR	Watersupply scripts	critical	8 Scripts to prepare and send ESS and ESP products			
ER	TAR	xmrg2grass	critical	Generates graphical display of data stored in xmrg files			
ER	TAR	Xsets scripts	critical				
ER	TIR	0Z_mapx_mat_Preprocessors	critical	runs NWSRFS preprocessors			
ER	TIR	12Zarchiveruns	important	runs additional NWSRFS runs			
ER	TIR	12Z_mapx_mat_Preprocessors	important	runs NWSRFS scripts			
ER	TIR	18Z_mapx_mat_Preprocessors	critical	runs NWSRFS data			
ER	TIR	3-way.awk	Nice To Have	Reformats Station Data from dd to dms			
ER	TIR	7amrec.pl	important	Gathers info for the phone recording			
ER	TIR	Add_contingency_QPF	important	runs NWSRFS contingency modes			
ER	TIR	Add_QPF	important	runs FMAP NWSRFS preprocessor			
ER	TIR	addZCZC	critical	adds ZCZC to products for AWIPS disseminations			
ER	TIR	alert.pl	important	Shows gage and product status			
ER	TIR	archive_FLDWAV_data	important	archive NWSRFS data			
ER	TIR	archive.mapxmat	important	runs NWSRFS archive data			
ER	TIR	archive_VERIFY_data	important	archives some verification data			
ER	TIR	areaDefStripS.pl	Nice To Have				
ER	TIR	AWIPS_bkupmove_dbtbls	critical	dumps various IHFS tables			
ER	TIR	AWIPS_bkupmove_files2erh	critical	sends various RFC backup data to ERH servers			
ER	TIR	AWIPS_bkupmove_files2ohrfc	critical	gets backup data from ERH data servers			
ER	TIR	batchpst_check	Nice To Have				
ER	TIR	bgn	Nice To Have	changes text colors			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	TIR	bgrannum	Nice To Have	generates random text colors			
ER	TIR	briefingflows	important	generates morning briefing values			
ER	TIR	card2norm.pl	critical	Converts NWSRFS normals to flat files			
ER	TIR	cfsd2cms.pl	important	converts cfsd to cfs			
ER	TIR	changehcltopound	critical				
ER	TIR	changeRC.pl	Nice To Have	removes forecast group IDs from rating curves			
ER	TIR	Check_QPF_Time	Nice To Have	checks date of QPF			
ER	TIR	clean_mm5	critical	removes old files from /data/fxa/Grid/LOCAL/netCDF			
ER	TIR	COERR2	critical	reformats and sends COE RR2 products			
ER	TIR	condense_coord	important	moves xmrg files to an archive directory			
ER	TIR	condense_hpc_qpf	important	sends HPC qpf files to an archive dirctory			
ER	TIR	condense_hpc_xmrg	important	sends HPC xmrg files to an archive directory			
ER	TIR	fgm.pl	critical	Shows status of RVF creation process			
ER	TIR	gmt*	important	26 scripts which generate various GMT maps			
ER	TIR	grd2*.pl	important	4 programs which generate various output from GMT grids			
ER	TIR	prepare*	critical	6 scripts that build and transmit AWIPS products			
ER	TIR	unique.pl	Nice To Have	removes duplicate lines from ppinit basin punches			
ER	TIR	Update0Z	critical	runs NWSRFS jobs for 0Z update			
ER	TIR	Update12Z	critical	runs NWSRFS jobs for 12Z update			
ER	TIR	update_ahps_prods	critical	runs NWSRFS jobs for ahps			
ER	TIR	UpdateSlidell	critical	runs NWSRFS jobs for morning RR5 and ffg transmission			
ER	TIR	vgf_backup.ksh	critical	ftps VGF files from HPC if necessary			
ER	TIR	webscript	important	sends HMD fimages to web			
ER	TIR	webscriptQPF	critical	sends QPF files to web			
ER	TIR	xdat	Nice To Have	meu script to run xdat			
ER	TIR	xmit	critical	copies files into transmission directories			
ER	TIR	xmit_ahps_fcst	important	sends AHPS files to CRH			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
ER	TIR	xmit_ffg	important	runs FFGxmit			
ER	TIR	xmrg2utm.pl	important	converts xmrg file to formatted text file			
ER	TIR	XMRGtoWeb	critical	Creates web graphics			
ER	TIR	xsets2opie.pl	Nice To Have	converts xsets output to another format			
ER	TIR	ZR.pl	critical	Creates HTML of current Z/R relationships			
SR	ALR	11 scripts like arc_shefd ...	important	Archives files on the RAX for WES SHARE			
SR	ALR	15 scripts like sju.makeB ...	important	Tars up MPE QPE files for archive on the RAX			
SR	ALR	16 scripts like launch_ar ...	important	Scripts to send Arcview generated images to WEB			
SR	ALR	2 files like ToNCDwr	important	Transmit ESP data to outside partners			
SR	ALR	2 scripts like biasreport	important	creates report of MPE biases			
SR	ALR	2 scripts like bias_reset.py	important	Allows the reset of mpe biases			
SR	ALR	2 scripts like dumppp24	important	creates daily rain gage report			
SR	ALR	2 scripts like monitor_ar ...	important	Script to analyze files archived on RAX			
SR	ALR	2 scripts like monitor_op ...	critical	monitor crtical scripts to make sure they are running			
SR	ALR	2 scripts like move_ohd_f ...	critical	Move files created by OHD scripts to the RAX			
SR	ALR	2 scripts like rcwhfs.tcl	important	Script used to update ratings in WHFS			
SR	ALR	2 scripts like Run_rejectdata	important	Creates shef coded product of rejected data			
SR	ALR	33 scripts like makeBMOSA ...	important	Scripts tar up QPE products for archiving			
SR	ALR	3 scripts like run_d2d	critical	Launches D2D			
SR	ALR	3 scripts like text_admin.tcl	critical	sends coordination messages			
SR	ALR	5 scripts like get_aws	important	Gets data files from sources outside of AWIPS			
SR	ALR	6 csh scripts like send1h ...	critical				
SR	ALR	6 scripts like sju.vl.scr	important	unloads postgres database tables for archiving			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	ALR	check_data_flow	important	Checks data flow			
SR	ALR	checkrvf	important	checks for missing RVF products			
SR	ALR	check_shef	critical	checks products that have shef coded errors			
SR	ALR	create_dump_sql	critical	Creates files for RFC backup			
SR	ALR	create_rc_trans	important	Creatings ratings files to be transmitted over SBN			
SR	ALR	dbpg	important	GUI interface to postgres database			
SR	ALR	end_ifp	critical	Ends hung ifp program			
SR	ALR	FilterFirstOrder.ksh	critical	Script to QC first order MTR data			
SR	ALR	formdata.sh	critical	creates forecast forms			
SR	ALR	get_aws	Nice To Have	Gets school network hourly shef coded precipitation data			
SR	ALR	get_hpcgrd	important	Gets QPF gridded files from HPC			
SR	ALR	getmap	critical	Creates MAP shef coded text products			
SR	ALR	get_ndfd	important	Gets NDFD data from HPC			
SR	ALR	getolddpas	important	Get missing DPAs			
SR	ALR	hdpradar.scr	important	unloads database table for archiving			
SR	ALR	linux_carry_cron	critical	Creates OFS carryover files			
SR	ALR	linux_clean_cron	critical	Purges local directory structure			
SR	ALR	linux_fcst_cron	critical	Creates OFS batch forecasts			
SR	ALR	linux_fldwav_cron	critical	Generates floodwav forecasts and carryover			
SR	ALR	linux_maprrs_cron	critical	Creates MAP and RRS Time Series for OFS			
SR	ALR	linux_xmaprrs	critical	Creates MAPX time series for OFS			
SR	ALR	location.scr	important	unloads a database table for archiving			
SR	ALR	lxbackup_cron.ax	important	backs up operational files under lx directory			
SR	ALR	makeCARRYOVERtar.sh	critical	creates tar file for archiving of OFS carryover files			
SR	ALR	make_gffg	critical	Creates gridded flash flood guidance products			
SR	ALR	makePRDtar.sh	critical	creates tar file for archiving of OFS PRD files			
SR	ALR	makeSTAGEirawtar.sh	critical	creates tar file for archiving of DPA data			
SR	ALR	movefilestosr	critical	moves files to SR server for backup system			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	ALR	NOS_fcst.ksh and NOS_Obs.ksh	important	Scripts sends FLDWAV data to NOS			
SR	ALR	pp24_shef	critical	Creates shef coded ATLHYDALR product			
SR	ALR	pr_rain	critical	Creates 24 hour rain gage shef product for PR			
SR	ALR	qps	critical	Creates QPF Mod for PR forecast area			
SR	ALR	qps_sum	important	Checks for QPF text products			
SR	ALR	radarloc.scr	important	unloads database table for archiving			
SR	ALR	rating	important	GUI to update ratings			
SR	ALR	rs	important	Script to send flatfiles to RAX			
SR	ALR	run_nohrsc	important	Programs used for basin delineation and uhg			
SR	ALR	run_rfo	critical	Allows the creation of ESG products			
SR	ALR	rvrwatch	important	monitor river forecast points			
SR	ALR	rwbiasdyn.scr	important	unloads database table for archiving			
SR	ALR	rwradarresult.scr	important	unloads database table for archiving			
SR	ALR	sac_states_to_pc	important	Sends data to LDAD for external SAS Display			
SR	ALR	send_24hrxmrg	Nice To Have	sends 24hr QPE xmrg files to WGRFC			
SR	ALR	send_apco	important	Transmit data to Alabama Power Company			
SR	ALR	send_cwil	important	Transmit ATLHYDALR product to Corp of Engineers			
SR	ALR	send_data_to_LDAD	important	Sends data to LDAD for external SAS Display			
SR	ALR	send_hurricane_ims	critical	Creates output for flood conditions map			
SR	ALR	send_rfc_products	critical	Sends text products out to the SBN			
SR	ALR	send_site	important	Sends QPF file for PR to LDAD			
SR	ALR	send_usgs	important	Transmits ATLHYDALR product to SRH servers			
SR	ALR	sju.movefilestosr	critical	moves files to SR server for backup system			
SR	ALR	sju.qpf	important	Displays QPF data for PR in XNAV			
SR	ALR	sloshdsp	important	Displays SLOSH data			
SR	ALR	tidedsp	important	Displays tide data			
SR	ALR	tsdata_save	important	Creates FLDWAV time series for archive			
SR	ALR	vl.scr	critical	unloads database table for archiving			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	ALR	xmit_atlrr2atr	important	Gets shef-coded data from Army Corps of Engineers			
SR	ALR	xmit_bhmrrmbhm	critical	Gets data from Army Corps of Engineers			
SR	ALR	xmit_lkof1-txt	critical	Gets shef coded data from Army Corps of Engineers			
SR	ALR	xmit_memrr1meg	critical	Gets shef coded data from Army Corp of Engineers			
SR	ALR	xmit_rdurrardu	critical	Gets shef coded data from Army Corps of Engineers			
SR	ALR	xmit_wwcg1	critical	Gets shef coded data from Georgia Power Company			
SR	ALR	xmrg_script	important	sends xmrg files to Army Corps of Engineers			
SR	FWR	alpha	important	manages and monitors client programs			
SR	FWR	arcEdit	Nice To Have	Edits archived observations			
SR	FWR	archive_reporter	important	archive checking utility			
SR	FWR	crestcat	critical	creates status map input file			
SR	FWR	critcheck	important	situational awareness tool			
SR	FWR	doAHPS	critical	gui to create AHPS probabilistic forecasts			
SR	FWR	dpaStat	important	Graphical / Text display of radar bias from mpe			
SR	FWR	fcst_Comments	critical	Adds comments to time-series forecasts			
SR	FWR	fopcontrol	important	resizes graphics using imagemagik			
SR	FWR	graphical hmd	critical	creates hydromet discussion (text & web version)			
SR	FW	ibwcrat	important	process IBWC rating tables			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
SR	FWR	local_nmap	critical	gui for operational qpf forecast process			
SR	FWR	make HCM	critical	creates hydro coordination message			
SR	FWR	makeHtml	important	Builds various html pages for web			
SR	FWR	makerecadvisory	critical	generates recreational river forecast			
SR	FWR	modClean	Nice To Have	Cleans old run-time mods from hydro model			
SR	FWR	nos_obs	critical	generates hydro forecast products for NOAA/NOS			
SR	FWR	primo	important	converts gridded xmrg files to text files for gis input			
SR	FWR	radarBias	important	displays radar bias information			
SR	FWR	rat2db	important	populates whfs rating table from ofs ratings			
SR	FWR	ratUtil	important	Edits and updates stage-discharge curves in ofs			
SR	FWR	rs	important	gathers precip data for input into gis			
SR	FWR	run_ofs	critical	front end for operational forecast model			
SR	FWR	shiftLog	critical	tracks and logs daily operational tasks			
SR	FWR	statusHtml	critical	creates html for status map			
SR	FWR	tclPrecip	critical	Creates operational precipitation products			
SR	FWR	tkDat	critical	displays operational time series			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	FWR	ups	critical	Transfers products to outside users via ftp			
SR	FWR	viewFOP	Nice To Have	views flood output product information			
SR	FWR	viewRVF	important	Graphical / Text display of issued river forecasts			
SR	FWR	wgSumTbl	critical	creates summary table for obs & fcst conditions			
SR	ORN	calbconvert	critical	program to convert archived data to NWSRFS datacard format			
SR	ORN	convertqps	critical	Sends qps product			
SR	ORN	dambrk_rules_of_thumb.py	critical	python tcl gui for dambreak analysis			
SR	ORN	fcstgen	critical	program to format forecasts to SHEF			
SR	ORN	ffg_create.lx	critical	tcl gui to run flash flood guidence			
SR	ORN	ffgParse.py	critical	ffg text formatting script for arcview/webpage			
SR	ORN	ffg_run.send.lx	critical	tcl gui to send flash flood guidence			
SR	ORN	fix_obs4status.py	critical	make corrctions to ihfs database for bad obs			
SR	ORN	fpmon.tk	important	system monitor			
SR	ORN	fsct_tools.tk	critical	menu to run many hydrologic applications			
SR	ORN	get_esp_input.lx	critical	gets data for ling range miss river forecast			
SR	ORN	get_latestobs.lx.psql	critical	gets stages for use with arcview/webpage			
SR	OR	getmap	critical	program to format maps to shef			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	N						
SR	ORN	get_map_data.lx	critical	gets map data for use with arcview			
SR	ORN	get_obs_with_cat.psql.py	critical	gets current stage flood status for web graphics			
SR	ORN	get_pc_ppdatanew.psql	critical	gets daily rainfall for use with arcview/webpage			
SR	ORN	get_tva_res.py	critical	gets tva data and formats to shef			
SR	ORN	main4	critical	formatter for NEWRVAORN product			
SR	ORN	movefilestosr	critical	moves necessary files for use with backup system			
SR	ORN	ofsshef	critical	program to format any nwsrfs data to shef			
SR	ORN	plotarc	critical	plots archived stage/map(x) data			
SR	ORN	process_products.tk.lx	important	decode text products for ofs input			
SR	ORN	ratshef	critical	converts stage to flow or visa versa			
SR	ORN	river summary	critical	program to format weekly river summary			
SR	ORN	riverwatch2home	critical	send certain text products to the riverwtach page			
SR	ORN	run_crestcat.lx	critical	gets forecast stage flood status for web graphics			
SR	ORN	runnmap_postprocess	critical	Creates qpf product and sends grib files			
SR	ORN	runnmap_postprocess_local	critical	Creates qpf product and sends grib files			
SR	ORN	send1hrqe00z	important	Sends old mpe gribs to users			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	ORN	send1hrqe06z	important	Sends old mpe gribs to users			
SR	ORN	send1hrqe12z	important	Sends old mpe gribs to users			
SR	ORN	send1hrqe19z	important	Sends old mpe gribs to users			
SR	ORN	send1hrqpe.sendold.etj	important	Sends old mpe gribs to users			
SR	ORN	send24hrqpe	important	Sends mpe gribs to users			
SR	ORN	send6hrqpe	important	Sends mpe gribs to users			
SR	ORN	sendold_qpf	important	Sends old qpf gribs to users			
SR	ORN	surge_menu	critical	menu for running surge models			
SR	TUA	admin.alarm	critical	rfc alarm script			
SR	TUA	arc_dumpgages	important	retrieve precip data from archive db for precip processing			
SR	TUA	arcfcstprog	critical	Displays historical forecasts and observations			
SR	TUA	arc_verify_obs	critical	move data from adb to abrfc verify db			
SR	TUA	auto_ffgalert	critical	automatically compare gridded precip to ffg and send out hcm			
SR	TUA	check_carrysave	important				
SR	TUA	check_fcsts	important				
SR	TUA	CheckHADS	important	alerts FIC if HADS data feed is down			
SR	TU	dcptime	important	determine when next data feed is for location			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
SR	TUA	delete_files	critical	deletes old files out of several dirs incl /data/fxa dirs			
SR	TUA	DisplayProcess3	critical	Precipitation processing algorithm - creates QPE			
SR	TUA	draw_webjpgs	critical	create river status graphics for web site			
SR	TUA	execfcstver	critical	Creates Southern Region Categorical Verification stats			
SR	TUA	fcstprog	important	Shows the last 5 days of fcsts with observations			
SR	TUA	ffg_alert	critical	suite of apps to compare gridded precip to ffg values and issue hcm for wfos			
SR	TUA	FGUS64-script	critical				
SR	TUA	ftp_w2k_dpas	critical	sends dpas from /data/fxa data dir structure to rfc backup system			
SR	TUA	gageQC	important	suite of apps to compare gridded rainfall to observed values			
SR	TUA	geo2shp	important	convert geodata files to shapefiles			
SR	TUA	GetOldDPAs	critical	backup method for retrieval of radar DPA files			
SR	TUA	GetProducts	critical	backup method to retrieve text and binary products			
SR	TUA	get_swe	Nice To Have	retrieve snow water equivalent data			
SR	TUA	getusgs	critical	backup method to retrieve DCP gage data			
SR	TUA	get-usgs-all	critical	get usgs data for awips and rfc backup system			
SR	TUA	get_vgfs	critical				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	TUA	gffg	critical	suite of apps to create gridded ffg products			
SR	TUA	goes24	critical	compute 24-hour precipitation totals from DCP gage sites			
SR	TUA	gxsets	critical	create hydrologic forecasts in SHEF			
SR	TUA	hads	important	goes to hads dir			
SR	TUA	hcm	important	Hydrometeorological Coordination Message formatter			
SR	TUA	Hmd	critical	creates online version of hydrometeorological discussion			
SR	TUA	HmdPrecip	important	formats 24-hour precipitation totals for web page			
SR	TUA	hyd	critical	format summary of 24-hour precipitation totals			
SR	TUA	hyd_int	critical	format summary of 24-hour precipitation totals for previous days			
SR	TUA	hydro	important	creates hydrographs for RFC webpage			
SR	TUA	idw	important	create xmrg from point data			
SR	TUA	LRProbFcst	critical	Long-range probability stage forecast formatter			
SR	TUA	meso24	critical	compute 24-hour precipitation totals from Oklahoma mesonet sites			
SR	TUA	monitor_fs	critical	check db for sites above flood			
SR	TUA	monitor_qc	important	check db for data failing qc			
SR	TUA	newfcstver	critical	Creates Southern Region Categorical Verification stats			
SR	TU	newxgif	critical	Creates images/netcdfs of hourly xmrg files			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
SR	TUA	nmap	critical	Program used by RFCs to create QPF			
SR	TUA	ofsmenu	important	create input files for nwsrfs and run ofs			
SR	TUA	okcmapxxx	critical	creates map product for dissemination			
SR	TUA	pcpnaccum	Nice To Have				
SR	TUA	phone	Nice To Have	ABRFC electronic phone book			
SR	TUA	ppmswe	important	processes monthly precipitation totals for water supply			
SR	TUA	PreProcess1	critical	Creates raw fields for DisplayProcess3			
SR	TUA	preprocessFFGGRIB_TUA.ksh	critical	sends out grib prods for rfc backup system			
SR	TUA	preprocessOUP.pl	critical	used to get products from ldad into awips for rfc backup			
SR	TUA	proc_wfo_ratings	important				
SR	TUA	qapLog	important	logging program for local daily product quality control			
SR	TUA	readgrd	critical				
SR	TUA	reject_data	critical				
SR	TUA	RFO	important	formats river flood outlook text products			
SR	TUA	rivstat	critical	suite of apps to create web status graphics			
SR	TUA	rr5-script	critical	keeps certain prods from going into shefdir /data/fxa/ispan/hydro			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	TUA	rr6-script	critical	keeps certain data from getting to shefdecode dir /data/fxa/ispan/hydro			
SR	TUA	sacco	critical	creates MODS file used by nwsrfs			
SR	TUA	sac_display	important	displays nwsrfs model sacramento contents			
SR	TUA	send_rfc/distributeProduc ...	critical				
SR	TUA	shefcheck	critical	checks shef data directory			
SR	TUA	shefdata	important				
SR	TUA	shiftLog	important	logging program for forecast shifts			
SR	TUA	show_zr	important	determine ZR mode of radars			
SR	TUA	snow	important	converts hourly snotel data to 24 hour total			
SR	TUA	stage_flow	important	converts stage data to discharge data and vice-versa			
SR	TUA	swsflow	important	Used in the statisical water supply project to drop data from rax			
SR	TUA	TV	Nice To Have	text forecast displayer			
SR	TUA	update_whfs	critical	insert new rating curves from ofs into ihfsdb			
SR	TUA	vgf	important				
SR	TUA	w2kshef	critical	sends shef prods to rfc backup sytem			
SR	TUA	xcentroid	important	determine precip centroid of map basin			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
SR	TUA	xdms_pg	critical	distributed model data viewer			
SR	TUA	xhdp_check	important	display radar hdp information			
SR	TUA	xmrg2kml	Nice To Have	convert precip to kml			
SR	TUA	xmrg2shp	critical	convert precip to shapefiles			
SR	TUA	xmrgutils	important	suite of apps to convert xmrg files to other formats			
SR	TUA	xsets_send scripts	critical	reconfigures and sends text prods			
WR	PTR	auto_daily_QC	critical	Generates precip and temperature inputs for hydrologic model			
WR	PTR	auto_specify	important				
WR	PTR	c5.sh	critical	process COE/BCHydro/BoR data			
WR	PTR	calc_filter	critical	modifies data to elimnate noise and bad values			
WR	PTR	calc_fq	important	calculates flood flow information for a site			
WR	PTR	calc_mean	critical	calcualtes mean flow values from observations			
WR	PTR	call.sh	critical	auto display of system failure/noticification			
WR	PTR	cbtt_shefencoder	critical	decode cbtt data and then encode into shef			
WR	PTR	cbtt_shefencoder.sh	critical	Encoder multi-agency data exchange to SHEF			
WR	PTR	COE Sets	critical	provides NWS hydrologic forecast in a format used by the COE			
WR	PT	Common libraries	critical	libraries for Daily QC programs			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
WR	PTR	dat_graph	critical	A visualization tool for viewing data			
WR	PTR	Day_list	important	provides data summary for data QC process			
WR	PTR	decode_nps_data	critical	decoder of COE data			
WR	PTR	discharge	critical	converts stage to discharge conversion			
WR	PTR	dmbrk_rot	important	dam break assistance program			
WR	PTR	dqc_bad.sh	critical	assemble bad precip sites and xmit from RFC to WFOs			
WR	PTR	dqc_parse_pcpn.sh	critical	prep precip for quality control/pull from DB			
WR	PTR	dqc_post_pcpn.sh	critical	post process quality controlled precip			
WR	PTR	dumpshef.sh	critical	utility to generate forecasts for xmission/generate model info for archive			
WR	PTR	fcst4plot	critical	creates input for GIS to generate of plots of forecasts			
WR	PTR	fmap.sh	critical	render mean area precip for hydro model			
WR	PTR	gage	important	displays rating and makes the conversion between stae and discharge			
WR	PTR	get_bchydro.sh	critical	SFTP data from Canada			
WR	PTR	get_brec.sh	critical	sftp data from the Bureau of Reclamation			
WR	PTR	get_c5.sh	critical	sftp data from Corps of Engineers			
WR	PT	get_ftp_files	critical	collects coop data			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
WR	PTR	get_nos.sh	critical	sftp data from the NOAA ocean service			
WR	PTR	get_ppl.sh	critical	sftp data from the Pacific Power & Light			
WR	PTR	get_pse.sh	critical	sftp data from the Puget Sound & Light			
WR	PTR	get_seattle.sh	critical	sftp data from the Seattle City Light			
WR	PTR	get_states	important	decodes dump of forecasts and model parameters			
WR	PTR	get_usgs.sh	critical	sftp data from USGS			
WR	PTR	grib	critical	encodes forecast data into grib format for dissemination			
WR	PTR	grib_xmrg	critical	Converts QC data to xmrg format			
WR	PTR	gsets	important	retrives forecasts from NWSRFS database			
WR	PTR	hmd.sh	important	generate/maintenance RFC hmd product			
WR	PTR	hpc_shefchange	critical	converts HPC QPF forecast for ingest into hydrologic models			
WR	PTR	ifx libraries	critical	database support libraries			
WR	PTR	ihfs_rate_load	critical	Loads and verifys rating table to ihfs database.. Used by WFOs			
WR	PTR	lib_tcls	critical	libraries for tcl programs .. these interface to database and create graphs			
WR	PTR	metar	critical	decodes metar messages and encodes them to shef			
WR	PTR	metar.sh	critical	process MTR data into SHEF code			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	PTR	MM	critical	QC of temp and precip data			
WR	PTR	mmstationlist.sh	critical	maintenance MM station list/QC process			
WR	PTR	monthlist	important	creates monthly data summary for water supply operations			
WR	PTR	nos_decoder	critical	decode NOS data and then encode to SHEF			
WR	PTR	ofs.archive.sh	critical				
WR	PTR	ofs.bx.sh	critical	NWSRFS batch forecast system (operational)			
WR	PTR	ofs.delmod.sh	critical	maintenance hydrologic model data			
WR	PTR	ofs.mods_helper.sh	important	auto generate NWSRFS mod data			
WR	PTR	ofs.mods_helper_tsoverlap.sh	important	NWSRFS mods overlap check			
WR	PTR	ofs.pp.d.sh	critical	process daemon for NWSRFS data transfer from postgres			
WR	PTR	ofs.pp.ofsde_perflog.sh	critical	reset postgres pointers for NWSRFS			
WR	PTR	ofs.pp.ofsde_recovery.sh	critical	recapture data from system fault for NWSRFS			
WR	PTR	ofs.pp.sh	critical	NWSRFS preprocessor executive			
WR	PTR	ofs.ssarreg_check.sh	critical	regulation date check for NWSRFS			
WR	PTR	ofs.ssarreg_update_batch.sh	critical	regulation update function for NWSRFS batch process			
WR	PTR	ofs.ssarreg_update_ifp.sh	critical	ssarreg update executive for NWSRFS IFP			
WR	PT	ofs.ssarreg_update.sh	critical	regulation update function for NWSRFS			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
WR	PTR	ofs.tempcheck.sh	critical	Observed temperature distribution function			
WR	PTR	ofs.tsoverlap.sh	important	NWSRFS mod overlap function			
WR	PTR	ofs.updown.sh	critical	hydrologic connectivity for autobatch process			
WR	PTR	ops.profile.sh	critical	establishes common system environment for local scripts/programs			
WR	PTR	ops.status.sh	critical	real time system/data process status for RFC			
WR	PTR	pa_verfiy	important	hydrologic verification			
WR	PTR	pcpnqc	important	Qc data sent to hydrologic models			
WR	PTR	peakflow	critical	read flow data and create peak flow determination			
WR	PTR	pgaccess	important	Forms program for interacting with the postgres database			
WR	PTR	ppl_shefencoder	critical	decodes PPL data and encodes into shef			
WR	PTR	ppmenu	critical	WS precip processor			
WR	PTR	ppnw	critical	WS precip processor/executable			
WR	PTR	pse_decoder	critical	decode pse data and encode into shef			
WR	PTR	qpf_monitor	important	monitor QPF vs observations			
WR	PTR	read_db	critical	driver pogram to get data from database feed QC programs			
WR	PTR	regrfs_espadp_xterm.sh	critical	joint model system xterm monitor			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	PTR	regrfs_esp_xterm.sh	critical	xterm monitor for NWSRFS ESP run			
WR	PTR	regrfs.sh	critical	Ineragency interface for shared NWSRFS modeling			
WR	PTR	regrfs_singleyear_ts.sh	critical	single year proces based on ensemble traces for flood control by fedeal agency			
WR	PTR	regrfs_stp_archive.sh	critical	single trace procedure archive			
WR	PTR	regrfs_stp_fs5bu.sh	critical	single trace procedure realtime backup			
WR	PTR	regrfs_stp_modsort.sh	critical	NWSRFS mod sort routine for single trace procedure			
WR	PTR	rfsld	critical	manages rating tables for all systems and programs			
WR	PTR	roo	Nice To Have	performs data backup functions for database			
WR	PTR	rvf_montior	important	montiors river forecasts and data			
WR	PTR	sbn_transfer	critical	sbn product router			
WR	PTR	schedule.sh	critical	RFC call back notification/HMD generation			
WR	PTR	seattle_coe_decoder	critical	decode scl data and encode into shef			
WR	PTR	sets	critical	encodes hydrologic forecasts into shef for dissemination			
WR	PTR	SETS	critical	A visualization tool for viewing data			
WR	PTR	sets.sh	critical	forecast generation/model output archive system interfaces with NWSRFS			
WR	PTR	skipkill.sh	critical	operational process monitor/interceptor			
WR	PT	snowmods.sh	critical	distribute snow mods to NWSRFS system			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
WR	PTR	snow_update	critical	update hydrologic snow parameters			
WR	PTR	stp_vs_obs	critical	review program for hydrologic forecasts			
WR	PTR	temp_qc	critical	qc observed temperature data			
WR	PTR	usgs_decoder	critical	decode usgs data and encode into shef			
WR	PTR	util	critical	support libraies for other programs			
WR	PTR	util.sh	critical	utility executive that performs >100 functions			
WR	PTR	util_tempbias.sh	critical	apply temp bias to forecasts			
WR	PTR	ws.archive.sh	critical	WS archive system			
WR	PTR	ws.awips_xmt.sh	critical	water supply product transmitter			
WR	PTR	ws.b500.sh	critical	WS file maintenance			
WR	PTR	wsbat	critical	WS executable			
WR	PTR	ws.batch.sh	critical	WS batch processor executive			
WR	PTR	ws.booklet.sh	critical	WS batch processor child			
WR	PTR	ws.cfile.sh	critical	WS characteristics processor/maintenance			
WR	PTR	ws.espcompare.sh	critical	WS regre/esp comparison function			
WR	PTR	ws.espdatabase.sh	critical	ESP database processor			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	PTR	ws.espweb.sh	critical	ESP web services processor			
WR	PTR	ws.fdatabase.sh	critical	ESP database poster			
WR	PTR	ws.frank.sh	critical	WS forecast rank processor			
WR	PTR	ws.midmonth.sh	critical	WS/ESP Midmonth Volume forecast aggregate			
WR	PTR	ws.orank.sh	critical	WS observed data rank processor			
WR	PTR	wsp	critical	water supply forecast model			
WR	PTR	ws.peakflow_get_volumes.sh	critical	WS regr/volume forecast extractor			
WR	PTR	ws.peakflow_observed_volu ...	critical	WS regr/peakflow verification			
WR	PTR	ws.peakflow.sh	critical	WS regr/peakflow processor			
WR	PTR	ws.plot.sh	critical	WS regr/plot preparation function			
WR	PTR	ws.ro_datarecovery.sh	critical	WS ro processor/data recovery mechanism			
WR	PTR	ws.ro_run.sh	critical	WS ro processor/child processor			
WR	PTR	ws.ro.sh	critical	WS ro processor/executive			
WR	PTR	ws.sh	critical	water supply forecast system executive			
WR	PTR	ws.verification.sh	critical	WS regr/verification system/public product generator			
WR	PTR	xffg.sh	critical	flash flood guidance processor			
WR	PT	xplots	important	suports maintenance for cross plot program			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	R						
WR	RSA	adj_cadwr_pillows	critical	reformats CADWR snow pillow data			
WR	RSA	adj_cadwr_ppm	critical	reformats cadwr monthly precip data			
WR	RSA	adj_cadwr_snow	critical	reformats cadwr snow course data			
WR	RSA	adj_cadwr_storage	critical	reformats cadwr reservoir storage data			
WR	RSA	arc_twxa	Nice To Have				
WR	RSA	AWIPS QPF Text Product Ge ...	critical	This script generates our QPF text products and sends them over the AWIPS SBN			
WR	RSA	call _twx	critical				
WR	RSA	checkafos.pl	critical				
WR	RSA	checkwest.pl	critical				
WR	RSA	chk_ta_l1	critical	quality control of hourly temperatures			
WR	RSA	chk_txtn_l1	critical	quality control of max/min temperatures			
WR	RSA	Climate Summary (Text Pro ...	critical	A collective of climate station precipitation (run daily)			
WR	RSA	copy_all_cpc	Nice To Have				
WR	RSA	create_hcm	critical	creates HCM product			
WR	RSA	create_hmd	critical	creates hmd product			
WR	RSA	create_hyd	critical	creates hyd product			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	RSA	create_obspcpn_24htot.pl	important				
WR	RSA	csv2oh.perl2_coe	Nice To Have				
WR	RSA	csv2oh.perl2_usgs	Nice To Have				
WR	RSA	csv2oh.perl3_moyr	important				
WR	RSA	DAILYQC	critical	PRECIP QUALITY CONTROL			
WR	RSA	Dambreak - includes seve ...	critical				
WR	RSA	dbserver_status	critical	checks the status of informix data servers			
WR	RSA	Discussion Template	critical				
WR	RSA	esp_scripts_gui.csh	important	sets up environment for esp_scripts.tk			
WR	RSA	esp_scripts.tk	important				
WR	RSA	ESP Trace Transfer for We ...	critical				
WR	RSA	fcstout_3	critical				
WR	RSA	fcstout_vfg	important				
WR	RSA	ffg_ffpinterp.tcl	critical				
WR	RSA	fill_SWE	important	preparation step for snow_updating			
WR	RSA	Flash Flood Guidance Graphics	critical	Creates flash flood guidance images using HRAP netCDF grids and the GFE			
WR	RS	Flood Outlook Product	critical				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
WR	RSA	Forecast Status Map (Fron ...	critical				
WR	RSA	Forecast Temperature Text ...	critical				
WR	RSA	fval_SUU.csh	critical	reads SUU data from fval table in fastetc			
WR	RSA	gen_hyd	critical				
WR	RSA	gen_hyd_arc	important				
WR	RSA	gen_twx	critical				
WR	RSA	getcelldata.tcl	important				
WR	RSA	ggtides	critical				
WR	RSA	Graphical River Forecast ...	critical				
WR	RSA	Graphical RVF Archive	critical				
WR	RSA	Historical Graphical Rive ...	critical				
WR	RSA	hydrodata	critical	creates data file for mapper program			
WR	RSA	ifpcom_manual	important				
WR	RSA	Lowflow	critical	Creates Delta Tide Forecast Bulletin			
WR	RSA	mapper	critical	displays hydromet data			
WR	RSA	mk_hyd	critical				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	RSA	mm2grib.tcl	critical				
WR	RSA	mm2rfs	critical	manages qpf and snow level forecasts for NWSRFS			
WR	RSA	mm_bas	critical	creates qpf products for basins			
WR	RSA	mm_pts	critical	creates qpf products for points			
WR	RSA	mm_spec	critical	creates input for mountain mapper specify			
WR	RSA	mod_killer	critical	removes obsolete MODS from NWSRFS			
WR	RSA	monthpx.pl	critical				
WR	RSA	Observed Precipitation an ...	critical				
WR	RSA	orog_avn_25pts	important				
WR	RSA	orog_avn_nrnca_text	important				
WR	RSA	orog_avn_srnca_text	important				
WR	RSA	orog_eta_25pts	important				
WR	RSA	orog_eta_nrnca_text	important				
WR	RSA	orog_eta_srnca_text	important				
WR	RSA	orog_mrf_day7_25pts	important				
WR	RSA	orog_mrf_days4-6_25pts	important				
WR	RS	orog_mrf_nrnca_text	important				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
WR	RSA	orog_mrf_srnca_text	important				
WR	RSA	Orographic Model	critical				
WR	RSA	PRD_archive	critical	ARCHIVES NWSRFS FS5FILES			
WR	RSA	precip_data	Nice To Have				
WR	RSA	Precipitation Summary	critical				
WR	RSA	Precipitation Verificiati ...	critical				
WR	RSA	print_twx	important				
WR	RSA	proc0	critical	computes period precipitation from incremental values			
WR	RSA	proc1	critical	computes hourly precipitation from incremental values			
WR	RSA	proc24	critical	computes 24-hourly precipitation from 6-hour values			
WR	RSA	proc6	critical	computes 6-hourly precipitation from hourly values			
WR	RSA	procM	critical	computes monthly precipitation from 24-hour values			
WR	RSA	QPE Graphics	critical				
WR	RSA	QPF Graphics	critical	Generates QPF graphics for web using the generation of a netCDF file and the GFE			
WR	RSA	QPF Organizer	important	Organized QPF points into a collective separated into WFOs			
WR	RSA	QPF Point Comparison	critical				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	RSA	qpf_scheduler.csh	critical				
WR	RSA	qpf_summary	critical	creates 6 day qpf summary product			
WR	RSA	r2c.tcl	critical				
WR	RSA	rawval_SUU.csh	critical				
WR	RSA	rdbhrly2oh.perl	Nice To Have				
WR	RSA	readindata_orog_00z.pl	important				
WR	RSA	readindata_orog_12z.pl	important				
WR	RSA	Reservoir Summary Text Pr ...	critical				
WR	RSA	River Summary Text Product	critical	River stage summary text product for the past 5 hours			
WR	RSA	Run_cpc_5day_all	important				
WR	RSA	Run_cpc_aj_all	important				
WR	RSA	run_ens_post_SAC_SJ	important				
WR	RSA	run_ens_post_SNOW	important				
WR	RSA	run_enspre_all	important				
WR	RSA	Run_error_5day_all	important				
WR	RSA	Run_error_aj_all	important				
WR	RS	Run_esp_5day_all	important				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
	A						
WR	RSA	Run_esp_aj_all	important				
WR	RSA	Run_esp_aj_gui	important				
WR	RSA	Run_esp_all_cpc	important				
WR	RSA	send_twx	critical				
WR	RSA	set_inp	critical				
WR	RSA	set_one	critical				
WR	RSA	set_twx	critical				
WR	RSA	seus_shef	important	preparation step for snow_updating			
WR	RSA	singstn	critical	displays time series data			
WR	RSA	Snowmelt Data Interface	critical	The interface the hydrologist uses to enter their snowmelt forecast.			
WR	RSA	snow_update	critical	updates snow water equivalent in models			
WR	RSA	SNOW UPDATE	critical	USED TO UPDATE NWSRFS SNOW PARAMETERS			
WR	RSA	start_monthly_precip	critical				
WR	RSA	start_wsf_entry	important				
WR	RSA	Temperature Plots	critical				
WR	RSA	tide2hour	important				

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	RSA	tide_reformat	important				
WR	RSA	track_qpf	important	creates qpf tracking products			
WR	RSA	twxa_arc	important				
WR	RSA	twxa_one	critical				
WR	RSA	twxa_xi	critical				
WR	RSA	twxa_xi_g	important				
WR	RSA	twxup_g	critical				
WR	RSA	twxup_one	critical				
WR	RSA	twxup_xi	critical				
WR	RSA	twxup_xi_edit	important				
WR	RSA	usgs_ratings	critical	updates/maintains river ratings			
WR	RSA	verify_hgf	important	stage forecast verification			
WR	RSA	verify_qpf	important	qpf verification			
WR	RSA	wsup_arc.pl	important				
WR	RSA	wsup_xmlencoder.tcl	critical	Creates xml file from water supply forecasts			
WR	STR	calculate data - 26 programs	critical	calculate data needed for operations			

Region	Site	Application Name	Importance	Function	NCRFC	ABRFC	NWRFC
WR	STR	create products - 8 programs	critical	create products			
WR	STR	format data - 17 programs	critical	format data for operational use			
WR	STR	modify files - 20 programs	critical	modify files for operational use			
WR	STR	move files - 12 programs	critical	move files			
WR	STR	nwsrfs interaction - 8 pr ...	critical	interact with NWSRFS			
WR	STR	operational help - 3 programs	critical	guide operational program flow			
WR	STR	parse data - 37 programs	critical	parse data from files for operational use			
WR	STR	quality control - 4 programs	critical	quality control operational data			
WR	STR	report/status of data - 1 ...	critical	create reports and monitor data			
WR	STR	Statistical Water Supply ...	critical	Statistical Water Supply package			
WR	STR	view data - 10 programs	critical	view operational data			

14. Acronyms

ATAN	AWIPS Technical Authorization Note
AWIPS	Advanced Weather Interactive Processing System
CAT	CHPS Acceleration Team
CHPS	Community Hydrologic Prediction System
CORBA	Common Object Request Broker Architecture ()
DHM	Distributed Hydrologic Modeling
FEWS	Flood Early Warning System
FTE	Full-Time Employee (or Full-Time Equivalent)
HCL	Hydrologic Control Language
HOSIP	Hydrologic Operations & Services Improvement Process
HSEB	Hydrologic Software Engineering Branch
HSMB	Hydrologic Science & Modeling Branch
IHFS	Integrated Hydrologic Forecasting System
IAO	International Activities Office
IM	Instant Messaging
NOHRSC	National Operational Hydrologic Remote Sensing Center
NWSEO	NWS Employees Organization
OMS	Object Modeling System
OSIP	Operations & Services Improvement Process
RAS	River Analysis System
REP	River Ensemble Processor
ResSim	Reservoir Simulation (model)
RSIS	RS Information Systems
RTi	Riverside Technology, inc
Sac-SMA	Sacramento Soil Moisture Accounting (model)
SOA	Service Oriented Architecture
TPS	Third Party Software
USACE	US Army Corps of Engineers
XEFS	eXperimental Ensemble Forecast System
XML	eXtensible Markup Language