Agile Product Lifecycle Management

ChangeCAST Developer Guide

v9.3.0.1

Part No. E15932-01
December 2009

Oracle Copyright

CONTENTS

Preface

The Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site http://www.oracle.com/technology/documentation/agile.html contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Note	To read the PDF files, you must use the free Adobe Acrobat Reader version 7.0 or later. This program can be downloaded from the Adobe Web site http://www.adobe.com.

The Oracle Technology Network (OTN) Web site http://www.oracle.com/technology/documentation/agile.html can be accessed through **Help** > **Manuals** in both Agile Web Client and Agile Java Client. If you need additional assistance or information, please contact My Oracle Support (https://support.oracle.com) for assistance.

Note	Before calling Oracle Support about a problem with an Agile PLM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Readme

Any last-minute information about Agile PLM can be found in the Readme file on the Oracle Technology Network (OTN) Web site http://www.oracle.com/technology/documentation/agile.html

Agile Training Aids

Go to the Oracle University Web page http://www.oracle.com/education/chooser/selectcountry_new.html for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Chapter 1

Introducing ChangeCAST

This chapter includes the following:

ChangeCAST is an easy-to-use Windows 2000 Server application for updating a target system with information from the Agile system. ChangeCAST creates transfer files, that an application adapter processes to send data from Agile to the target system tables.

ChangeCAST allows a target system to take advantage of Agile's straightforward ECO/MCO and BOM management capabilities, together with the target system's manufacturing and planning tools.

ChangeCAST Data Transfer

ChangeCAST produces a transfer file, which is a flat ASCII file containing information pertinent to change orders that have been created and released in Agile. ChangeCAST uses a profile to identify the target system's attributes and define the Agile-to-target mapping used to format the transfer files. The profile also defines the transfer schedule, the output protocol, the destination location of transferred files, and email notification.

The application adapter can be an unattended utility that imports a flat ASCII file, performs robust validations, and executes database updates or updates via the target system API. It must provide transaction logging, clear and concise error reporting, and measures to ensure that invalid data does not pass to the target database. It must also do some minimal file management.

In addition, the application adapter might need to qualify the data, and to reformat it as required by an existing import utility, if you cannot directly generate the required input with the ChangeCAST profile.

The transfer process is one way, from the Agile system to the target system.

Integration Checklist

The integration process includes the following steps. Each step is the subject of one of the later chapters.

- □ Gather and document the initial requirements.
- □ Draft the application adapter functional and program specifications.
- □ Update the Agile Web Client settings.
- □ Create the ChangeCAST profile.
- □ Develop a data processor (where needed).

- Develop the application adapter.
- Perform QA unit testing.
- Perform end-to-end (full system integration) testing with the user, to validate all business processes are functional.

Working Outline of the Integration process

You can develop a working outline along these lines in the integration process.

1. Gather and document the requirements.
    - Customer usage requirements
        - Current (pre-Agile)
        - Planned (post-Agile)
    - Target system requirements
        - Current (pre-Agile)
        - Planned (post-Agile)
    - Agile system requirements
2. Draft functional specification for baseline application adapter.
    - Receive signoff from Agile users, target system users, and anyone administering the integrated system.
3. Draft design/program specification for baseline application adapter.
    - Receive signoff from Agile users, target system users, and anyone administering the integrated system.
4. Configure/customize Agile environment for baseline application adapter.
    - Agile Web Client
    - ChangeCAST
5. Develop schedule and identify all users required to participate in full system integration testing.
6. Configure target system/Agile environment for customized application adapter (on non-production environments of both Agile and the target system).
    - Agile Web Client
    - ChangeCAST
    - Target system
7. Develop test plan for Agile to target system testing.
    - Software unit tests
    - System functional tests
8. Code, test, debug, and rework.
9. Develop and implement training schedule.
10. Document support.

Chapter 2

Systems Integration Analysis

This chapter includes the following:

System analysis is the key to successful integration between any two systems. Early identification of the affected processes during integration will save time in the development of an adapter. This chapter examines the analysis you need to perform before creating an application adapter.

Overview

As the integration developer, you need to understand the company’s business process, and use care and planning throughout the integration. You need to decide where data should be managed—in Agile or in the target system:

- Spend ample time gathering the user requirements; define an application adapter that reflects the best use of the Agile and target systems.
- Review the target system programs for the subroutines that sort, validate, and import records. You should be familiar with the standard target system routines for these tasks.
- If the target system does not have accessible subroutines, decide how to access the target system tables, import, and write the validation code.
- If you are developing an application adapter for more than one version of the target system, compare the routines you use for each version. Ensure to consider fundamental differences and document deviations.

Field Ownership

To integrate the Agile and target systems, define:

- data that will be managed in the target system only
- data that will be managed in Agile only, and
- data that will be shared by being entered in Agile and transferred to the target system.

Understanding the ownership will help you define the behavior and data management of additions, changes, and deletions between the systems. Assign an owner—Agile or the target system—for each data field. The owner will be able to overwrite any data in that field.

EXAMPLE

You need to manage effective dates in the target system. The engineer using Agile may suggest a start effective date, or you may send a “default” to the target system for each new item, but the target system should be the owner of all effective date fields. After you add a new item, the planning staff will set the real effective date. In this case, a change order from Agile must not modify the effective date, and the application adapter must not pass the original Agile effective date in a change order record.

Field Worksheets

After establishing how Agile and the target system work together, identify the field lengths, validations, formats, cases allowed, keys, and dependencies. You can find Worksheets that identify the attributes and properties in Appendix B.

For the fields that Agile is to own, mark up the worksheet indicating modification to field name, validations (list population), user interface (format modifications), field length, whether the field is required, and the case allowed. Identify the behavior of additions, changes, and deletions behavior.

Similarly, for the target system, use the worksheets to identify all tables and fields affected by the interface, and specify ownership, validations, format, and so on.

Database Relationships

It is important to note the interdependencies among a system’s databases. In most systems, parts or items must exist in a master data set before they can be used on a bill of material. Document these dependencies and how application adapter can handle them.

Consider the following questions during your system analysis:

- What tables or data sets must have an entry before you can enter a record for a part, bill of material, or change order? Is this an automated or manual process?
- Can you load items, dynamically in the target system as a function of creating or modifying the bill of material?
- Does the target system have a simulated data load or interactive import utility that the adapter can use to drive the Agile data into?
- Of the required fields for the identified affected tables or data sets, which have values that you must validate against another data set or table upon or prior to entry?
- Are there fields populated in the integration that are based on values of other fields within the same record? A similar record in another data set or table? Common, system, or programmatic variables or tables?
- Will change orders require that ECO records be transferred to the target system because of database dependencies or usage by the user?
- Are the item and BOM attributes correctly added or updated?
- What attributes or tables need to have information so that this part or assembly can be:
    - Purchased
    - Sold

  - Shipped
  - Returned
  - Built
  - Stocked
  - Accounted for
  - Planned
  - Reported
  - Costed

- Do you need to populate the affected tables immediately by the application adapter, or can you populate them on an as-needed basis?
- Are there server or database configurations that you need to consider to enable the target system for this interface (such as FTP)?

Target System

You need a full understanding of the target system issues as you develop the application adapter, for example:

- What combination of fields do you use to make a BOM row unique? How will this affect the information that transfers from Agile?
- How are dates for effective and obsolete BOM rows used? Are BOM rows entered initially with a zero date (00/00/0000) or can you use the effective dates from the ECO?
- Data from Agile includes 4-digit years; how will the target system handle these dates?
- Does the target system operating system have an FTP file size or line length limitation?
- How do you store reference designators individually?
  - In text fields of *nn* characters each, sequentially oriented by a reference designator field counter?
  - Are there character limits or record limits?
- How do you make changes to reference designators?
  - Delete and update?
  - Obsolete the old record using an obsolete date with the old data and add a new record with a new data?
- Are reference designators required in the target system when the integrated Agile/target system is in place?
- Are full BOM configurations kept for revision history, or are just the deltas kept?
  - When you update a BOM, is a copy of the entire BOM retained for history, or are deltas maintained at a component level only?
  - Are “net” changes required only?
- How are BOM revisions stored and maintained?

  - At the BOM component level?
  - At the assembly level?
- Are component or key field changes dealt with differently from non-component, non-key field changes?
- How would a change to a BOM row be done interactively? How would that data be stored in the BOM table?
- Is there a need or ability to view historical BOM configurations on the target system?
- How does the target system handle substitute or alternate parts?
- Does the target system have an ECO module? If so, is this the best place to integrate Agile ECO data with the target system?
- Can field-, user-, and screen-level security be instituted in the target system to avoid inadvertent updates to the databases?
- In what order should additions/changes/deletions be done at the assembly level?
- In what order should additions/changes/deletions be done at the BOM component part level?
- How will MCOs be handled compared with ECOs?

The design of the application adapter should explain the target system’s BOM behavior and change logic. Consideration of these questions will help you design a robust application adapter.

Chapter 3

Application Adapter Requirements

This chapter includes the following:

In developing a design specification for the application adapter, you will generate information needed to configure Agile Web Client and ChangeCAST. You will also develop an implementation plan for integration with Agile software that addresses the following issues:

- How to achieve functional system integration between Agile and the target system
- How to establish error diagnosis, handling, and troubleshooting
- What report utilities to develop
- What documentation to provide (user and system administrator)
- The type of support to provide
- How to upgrade the application adapter if the target system is upgraded

You will also develop information from which a certification plan can be created.

Overview

Depending on the target system's capabilities and business needs, the complexity of the application adapter may vary. Most application adapters perform some or all of these functions:

1. Retrieve a transfer file and parse the data to evaluate the Item and BOM additions, changes, or deletions (where supported), in chronological order. (Required.)
2. Assign record IDs, sort the records, or otherwise prepare the data for importing. You can manage this in a temporary text file or interim table, depending on the application adapter design.

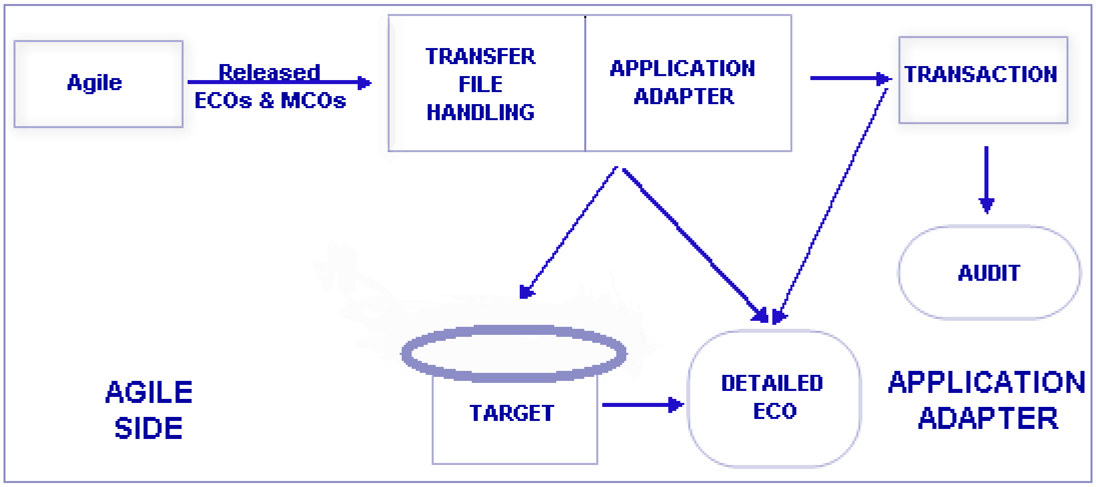
   Note Agile maintains a unique ID for every BOM row that you add, change, or delete. To ensure synchronization, we recommend that you use this ID to match the target system's BOM data.

3. Validate the additions, changes, and deletions, using routines that are identical or equivalent to those used by the target system for interactive transactions. (Required.)
4. Upon change order validation, write the data to the target system tables, and commit the

transaction. (Required.)

5. Move the processed transfer file to an archive directory or location, or mark it as completed on the interface tables. (Required.)
6. Stop processing if you find any error during change order validation, and ensure that no record from the transfer file is committed.
7. Write error messages to an error log file, and move failed transfer files to an "error" directory or location. (Required.)
8. Notify the user of the error, ideally via email to a user identified in a configuration or setup file. The user must release a new change order to correct errors, according to a predefined application adapter procedure.
9. Write all transaction information (pass and fail) to a transaction log file. (Required.)

Processes of the application adapter



Note The transfer process is one-way, from Agile to the target system.

Use Agile to create Parts and BOMs and maintain change order history. To avoid confusion and data entry issues the owner must identify and maintain fields. For details, see Field Ownership on page 3.

General Design Considerations

Consider the following issues while designing the application adapter.

- **24/7 operation —** Most corporations require that any application that works with enterprise data be capable of operating round the clock, without failure or user intervention.
- **User login —** The application adapter should not require that a specific user log in to operate. Some organizations may want different users to access different application adapter processes or logs.
- **Security —** The application adapter must not compromise target system security. The validation routines that the application adapter automates must be as secure as those in the manual data

input environment.

- **Year 2000 (Y2K) support —** The application adapter needs to manage century data in the *21*st (20*nn*) century. Starting with Release 5, the year is always in 4-digit format, so the application adapter must anticipate this format in the transfer file.
- **Internationalization —** The application adapter may need to manage extended ASCII characters, or be locale-aware. Discuss this with your target system team, so expectations are correctly set.
- **ISO and GMP compliance —** The application adapter process may need to be tested and documented to support ISO certification. For medical device manufacturing, this may mean retesting for GMP certification. Check with the Quality Assurance team, or others managing the ISO or GMP processes.
- **Affected and unaffected databases —** Make sure to identify databases that need manipulating. Ensure to document related but unaffected databases too.
- **Level of rollback supported —** Consider that no change order should be committed if it has the possibility of corrupting the database. Minimally, a mid-processing rollback must be supported. Depending on the database update and backout capabilities, this may mean that a change order and all the related items are fully validated and then committed (or, if not fully validated, not committed), or that there is the ability to partially commit data but back it out if there is a failure to commit. Providing the ability to back out a successfully committed change order is not recommended, and has severe drawbacks unless extensive precautions (with all the reverse validations) are made. Care should be taken to document the process of how a backout or rollback will be performed. If an automatic rollback process fails, manual procedures must be deployed and documented to ensure that erroneous data does not remain in the database.
- **Limitations of the application adapter —** Itemize any and all known areas that the application adapter will not address either by design or by oversight, as it becomes necessary.
- **Customizing layer —** If the user can access a custom layer of code that is built, document it completely, and describe the coding requirements for access.

Target System Interface

When designing the interface with the target system, you must consider the impact to the system. Minimal impact to the target system should be a rule of thumb. Things to consider include:

- What is the deliverable? If possible, deliver executable code only. This typically takes up less space and enables users without compilers to use the code. The code should be designed and structured for speed and transparency to the user.
- Will the application adapter reside on the same system as the target system? Or the Agile system? If so, how will this impact the target or Agile system? If not, what are the hardware requirements and the remote validation processes?
- Will an interface table be placed on the target system to enable the integration? What will this table look like? What is the capacity of this table? What maintenance is required?
- Will default tables, records, or templates be used for the application adapter to deal with loading default data for initial adds? Changes?
- Will a separate database or log be used for transaction logging to enable error, status and audit

reporting?

- Will additional locations be used for maintenance of the transfer files? Where will these be located, and how will they be maintained?
- Does the user have additional programs (security, database maintenance, and so on) running that were not considered in the test environment? Document how these might affect the application adapter's performance.

Components and Issues

When designing the application adapter, you need to consider the following:

- **Installation routines or programs** — an application adapter installation procedure can be as simple as copying the files to the user's system where the application adapter will reside. Or it may use an entirely automated, industry-standard or platform-specific installation program. Whichever method you choose, ensure that you install files in the correct locations. This will make documentation, use, maintenance, upgrades, and support easier and readily reproducible, if required. If you must modify the system access or capabilities to enable use of the application adapter, state this as a part of the installation.
- **Transfer file location** — identify a single location as an inbox for the ChangeCAST transfer files that the application adapter is to process. ChangeCAST allows the administrator to select the protocol for the files to be deposited either locally or as a result of an FTP event. Because you set-up ChangeCAST first and then launch it as a service, the protocol will not usually change after you test and deem it successful. For this reason, it is best to identify a deposit location from the outset. Whether the transfer file is targeted for a local directory or an area on a host system, the application adapter should be able to retrieve the file for processing without manual intervention.
- **Parsing** — the transfer file generated from ChangeCAST is an ASCII file. Parsing the transfer file may require field level validations. Define the module that parses the file, and what temporary files or tables, if any, are used to identify or sequence information prior to validation.
- **Conversions** — may require converting data prior to validation. For example, dates in the transfer file are in a human-readable format that has been previously set in the ChangeCAST profile (with ChangeCAST 5.0 and later, date output is in four-digit format). If the target system database requires dates to be stored as an integer, conversion will be required. In this case, date conversion routines would be typically exposed in a target system due to the fact that dates must be converted for input and update purposes. The application adapter should use the standard conversion routine.
- **Validation** — perform field-and record-level validation on all appropriate fields. Handle any and all validation possible in Agile through Agile Web Client settings; to ensure that you validate data at the time of input or release.

  Any field that is required by the target system must be checked for correct type and the presence of data, or it must be filled with a default value by the application adapter. Validate fields and records for format and content.

  Determine what the application adapter does with the transfer file data to ensure that it is valid for the target system. Identify the subroutines, libraries, APIs, or other code that will be used for each type of data validation. If you document them in standard target system references, include pointers to those references, or paraphrase key text.

- **Database Updates** — validate each change order you send to the application adapter as a complete set of records. If any data within the change order is in error, the entire change order is invalid, and no data should update the target system database. The application adapter must stop processing, write an error message to a transaction log, and notify the user of the error. Consider the following questions about the add and update logic:
  - At what point does the application adapter update the affected tables?
  - Are there database utilities that support data rollback or backout?
  - At what level (record, table, change order) will the transaction log be updated?
  - How will the update affect the users? Other records in the table? Other tables in the database? In other words, what level of locking will be deployed, and what performance issues can be expected?
  - Even though the change order may be validated perfectly, a transaction update may be interrupted due to power loss or time out. Will the application adapter be able to handle these end cases? How?
  - In what order will additions, changes, and deletions be done for parts? BOMS? Why?
  - How can questionable change orders be evaluated prior to being committed to the target system database? The design should include a validation-only option.

Defining Unit Tests

The unit tests should cover:

- Installation and configuration components, to ensure that they are complete and functional.
- Each field in each of the affected databases. Test for length, case, format, match in a lookup table or database (where applicable), presence if required, field additions, field changes, and filling with a default. Each field should be tested with data that will generate positive and negative results.
- Each record's proper population of the database.
- Updates to a table and the impact on related tables, rows, fields.
- Tests for changes when an incoming field is longer than, shorter than, or not changed from the original field.
- Database presence or availability. For instance, will the Item Master be locked at any time during order entry? Will the BOM tables be locked? Test all cases of database availability:
  - All databases or tables available
  - Some databases or tables not available
  - No databases or tables available
- Change order–specific re-transmission.
- Graceful exit, error messages generated, and no impact to the database in the event a change order is not validated or cannot be processed.
- File management during and after successful or failed processing.
- Interactive or automated processing of multiple transfer files with individualized change order

error handling.

- Any switches or options the application adapter offers that may affect the contents of the transfer file or the manner in which data is processed.

See System Testing on page 47 for details on testing.

Processing

Ultimately, the application adapter components need to be put into a cohesive interface for the user. This will provide the regular process by which a change order is transferred to the target system. This process should consider that the user may have specific requirements or reasons for processing change orders in a certain way, such as the need to:

- Manually process a single transfer file with the application adapter interactively, to view messages in real time
- Manually process a single transfer file out of sequence with the application adapter, to accommodate a user-specified processing sequence
- Manually process a range of transfer files with the application adapter, with the ability to select processing behavior (stop, continue on error, and so on) in the event of a problem with a change order
- Automatically process all transfer files in the queue location on a scheduled basis, with ability to stop all or some processing in the event of a problem with a change order

Due to the nature of change order processing and the interdependencies that can exist between one change order and the next in the queue, single-thread processing of change orders is the preferred processing method. It is imperative that any multiple-file processing consider each transfer file individually and thoroughly before the next transfer file is processed. The files are numbered in the order the change orders are released, and they should be processed in that order. The file numbering convention can be defined by settings in the ChangeCAST profile.

You must consider environments where multiple changes to a BOM or part may be released in a single processing session. In this case, if the first update fails, what will be done with the subsequent change?

File Management

As a transfer file is processed, it is a good idea to move it from the ChangeCAST FTP or copy destination to a working location. Once you process the transfer file, the application adapter should move it to a “completed” or “error” location depending on the success or failure of processing the file. The location names and handling described here are suggestions only. Relevant file handling must be described and addressed in the application adapter requirements.

Documentation

The following sections discuss subjects that you need to document to aid in the support and use of the application adapter. Also, see Agile Administrator Settings on page 19.

Installation

Make sure that the installation guide covers not only the tasks involved in installing the application adapter but also the tasks involved in setting up Agile to work properly with the application adapter.

Cover each of the appropriate areas, being specific as to what must be done and why.

- Identify and configure required Agile fields.
- Set up Agile SmartRules.
- Identify and configure Page Two attributes to be used, and set up the Page Two form.
- If appropriate, change roles and privileges to allow or disallow changes to identified fields after release and disallow unreleasing of change orders.
- Configure the ChangeCAST profile.

While you may use an automated installation for copying the programs onto the target system, it is a good idea to document the steps required for the installation to be successful. The steps you document might include the following:

- Decide whether ChangeCAST is to reside on the same server as Agile, or a different server (for performance reasons).
- Prepare the system. Make the working, complete, and error transfer file locations, and create a place for the application adapter log to reside, depending on how your adapter will process from these locations.
- Set up users.
- Set up any custom requirements.
- Set up the interim database (if necessary).
- Set up customer defined field defaults.
- Make any modifications required to accommodate the customer environment.
- Load the code. Compile it, if you are delivering source code, to make sure that it will recompile if the executable version is deleted, and to determine if it must be modified or linked to customer-specific libraries.
- Test to ensure that the environment is set up correctly and to check if you can execute the code.

Basic Use

This document will be used as a reference guide for every user of the application adapter. It should discuss how to use the application adapter and describe any of the daily issues a user might encounter. Include an overview of how a change order is processed from start to finish (Agile to target system).

Consider including these items in a user-oriented guide:

- Process diagram
- Field mapping
- ChangeCAST scheduling
- Manual application adapter processing
- Automatic and scheduled application adapter processing
- Application adapter processing limitations
- Common questions and answers
- Location of transaction logs, working, complete, and transfer files
- How to print or view Error, Audit, and change order Status reports

Field Mapping Document

Provide a field-to-field mapping document as a means to do the following:

- Validate that specific fields in Agile will be mapped to the appropriate field on the target system
- Troubleshoot ChangeCAST problems

You can generate the mapping document by obtaining a hard copy of the Mappings report from the ChangeCAST profile to be delivered with the application adapter. As the report is in a very simple text format that is not aligned, open the file in an Excel spreadsheet, using the “=” as the delimiter to create a columnar mapping document.

Target System fields	Agile fields
[PART.PART_NUM Length:30]	[Parts.General Item Attributes.Number:1001]
[PART.PART_DESC Length:100]	[Parts.General Item Attributes.Description:1002]
[PART.PART_SIZE Length:1]	[Parts.General Item Attributes.Size:1068]
[PART.PART_LINE Length:20]	[Parts.General Item Attributes.Product Lines:1004]
[PART.PART_PHASE Length:20]	[Parts.General Item Attributes.Lifecycle Phase:1084]
[PART.PART_REV Length:20]	[Parts.General Item Attributes.Rev:1014]
[PART.PART_INC_DATE Length:9]	[Parts.General Item Attributes.Rev Incorp Date:1017]
[PART.PART_REL_DATE Length:9]	[Parts.General Item Attributes.Rev Release Date:1016]
[PART.PART_TYPE Length:9]	[Parts.General Item Attributes.Part Type:1081]
[PART.PART_CATEGORY Length:20]	[Parts.General Item Attributes.Item Category:1082]
[BOM.PARENT_NUM Length:30]	[Parts.General Item Attributes.Number:1001]

Target System fields	Agile fields
[BOM.CHILD_NUM Length:30]	[Parts.BOM Table.Item Number:1011]
[BOM.FIND_NUM Length:9]	[Parts.BOM Table.Find Num:1012]
[BOM.QPA Length:5]	[Parts.BOM Table.Qty:1035]
[BOM.EFFECTIVE_DATE Length:9]	[Change Orders.Affected Items Table.Effective Date:1079]
[BOM.OBSOLETE_DATE Length:9]	[Change Orders.Affected Items Table.Obsolete Date:1078]
[BOM.REF_DESIG Length:1024]	[Parts.BOM Table.Ref Des:1019]
[BOM.REMARKS Length:512]	[Parts.BOM Table.Notes:1036]
[ECO.CHANGE_STATUS Length:20]	[Change Orders.General Change Attributes.Status:1030]
[ECO.CHANGE_NUMBER Length:30]	[Change Orders.General Change Attributes.Number:1047]
[ECO.CHANGE_CATEGORY Length:30]	[Change Orders.General Change Attributes.Change Category:1060]
[ECO.REASON_CODE Length:30]	[Change Orders.General Change Attributes.Reason Code:1049]
[ECO.CHANGE_ORIGINATOR Length:100]	[Change Orders.General Change Attributes.Originator:1050]
[ECO.CHANGE_ ADMINISTRATOR Length:100]	[Change Orders.General Change Attributes.Change Analyst:1099]
[ECO.DATE_RELEASED Length:9]	[Change Orders.General Change Attributes.Date Released:1051]
[ECO.CHANGE_DESCRIPTION Length:1023]	[Change Orders.General Change Attributes.Description of Change:1052]
[ECO.CHANGE_REASON Length:1023]	[Change Orders.General Change Attributes.Reason For Change:1053]
[ECO.CHANGE_DATE Length:9]	[Change Orders.General Change Attributes.Date Originated:1061]
[ECO.CHANGE_TYPE Length:30]	[Change Orders.General Change Attributes.Change Type:1069]
[ECO.CHANGE_PRODUCT_LINE Length:20]	[Change Orders.General Change Attributes.Product Lines:1003]
[ECOITEMS.CHANGE_NUMBER Length:30]	[Change Orders.General Change Attributes.Number:1047]
[ECOITEMS.ITEM_NUMBER Length:20]	[Change Orders.Affected Items Table.Item Number:1054]
[ECOITEMS.OLD_REV Length:20]	[Change Orders.Affected Items Table.Old Rev:1055]
[ECOITEMS.NEW_REV Length:20]	[Change Orders.Affected Items Table.New Rev:1056]

Target System fields	Agile fields
[ECOITEMS.LIFECYCLE_PHASE Length:20]	[Change Orders.Affected Items Table.Lifecycle Phase:1057]
[ECOITEMS.CHANGE_ FUNCTION Length:20]	[Change Orders.Affected Items Table.Change Function:1058]
[ECOITEMS.DESCRIPTION Length:100]	[Change Orders.Affected Items Table.Description:1059]
[ECOITEMS.EFFECTIVE_DATE Length:9]	[Change Orders.Affected Items Table.Effective Date:1079]
[ECOITEMS.OBSOLETE_DATE Length:9]	[Change Orders.Affected Items Table.Obsolete Date:1078]
[ECOITEMS.ON_ORDER Length:10]	[Change Orders.Affected Items Table.On Order:1085]
[ECOITEMS.STOCK Length:10]	[Change Orders.Affected Items Table.Stock:1086]
[ECOITEMS.WIP Length:10]	[Change Orders.Affected Items Table.Work In Progress:1087]
[ECOITEMS.FINISHED_GOODS Length:10]	[Change Orders.Affected Items Table.Finished Goods:1088]
[ECOITEMS.FIELD Length:10]	[Change Orders.Affected Items Table.Field:1089]

Error Handling and Troubleshooting

The specification should clearly document what happens if you detect an error in the transfer file.

Error Notification

When an error occurs, the application adapter should notify a specified user, or group of users, typically via email. The users can be defined during the application adapter installation, read from a target system table or configuration script, or defined in a network alias. Design the application adapter so different types of messages will alert different users, or use different messaging systems.

Use this section of the specification to describe how notification works in the application adapter. For example, describe what processing can be automated if the email reaches the user successfully, and what occurs if the email message fails.

Error Recovery

The application adapter should stop on the first error it finds in any change order. Because any sequence of change orders may have interdependencies, the failed change order may need to be corrected before processing continues. Provide a detailed set of instructions on how to recover from an error, whether manually or by automated process. These steps should discuss diagnosis of a problem and appropriate measures for resolution.

The user should create a corrected change order and release it in Agile Web Client. ChangeCAST will process the new information and send it to the application adapter. If not all errors in the change order are corrected, the application adapter will stop again, requiring a second correction, and so on.

To minimize the need for user error management, the application adapter can use a temporary table to store all records in the change order during validation, and attempt to validate the entire transaction set before passing or failing the change order. Then a complete set of errors for a given change order can be reported for appropriate corrective action.

Troubleshooting

Determining the root cause of a change order error must be done in a systematic way. Design a step-by-step process to compare old Agile part and BOM configurations with current part and BOM configurations in the target system and the transfer file. Because each of these components is a key factor in the successful transfer of data from Agile to the target system, each must be considered when the processing of a change order fails.

Here are some areas to consider when developing a troubleshooting methodology:

- Does the shared data of the prior revision of the part or BOM in Agile look exactly like the current revision of the part or BOM on the target system? Identify all differences and similarities. Identify which system is correct and make appropriate changes to the incorrect database to ensure that data will be updated with the next attempt.
- In Agile, are there effective dates on all affected items? Take corrective action if necessary.
- In Agile, are there new revisions on all affected items? Take corrective action if necessary.
- In Agile, have all affected items been placed on the **Affected** Items tab? Take corrective action if necessary.
- In Agile, are all required fields filled in? Take corrective action if necessary. (This may require the Agile administrator to make changes to ensure that this problem does not recur.)
- Does the transfer file correctly reflect any changes to parts and old versus new BOM configurations? Check the ChangeCAST profile.
- Does the transfer file have the correct mapping of data? Check the ChangeCAST profile.
- Does ChangeCAST generate the file in the correct format? Check the ChangeCAST profile.
- Does the transfer file use correct conversion logic for dates, nulls, CR/LF, and so on? Check the ChangeCAST profile.
- Have commands to add, modify, and delete parts or BOMs been disabled, or has access been changed on the host system? See the target system or database administrator.

Chapter 4

Agile Administrator Settings

This chapter includes the following:

The Agile database is delivered with settings for field names, lengths, and formats that may or may not match those of the target system. The Agile database settings are controlled by the Administrator settings, which are a comprehensive array of database objects, classes, roles, preferences, and so forth. The Administrator settings allow modification of field names, lengths, and types to match those of the target system, although data already entered in the Agile database is not affected by changes in field length.

The Administrator functions are set within the Java client, and are only available to Agile users who have been assigned the Administrator privilege. Prerequisite to creating the application adapter:

- A thorough knowledge of Administrator; for details, see the *Agile PLM Administrator Guide*.
- A thorough understanding of what data is to be maintained within and transferred from the Agile system.
- Meet with key end-users to determine their needs and to create a clear specification for the integration process.

When you are clear about what you need the Agile system to do, you (or the Agile administrator) will populate validation lists to match those in a target system table to ensure that only valid entries will be passed to the target system. You will identify fields that require data, forcing them to be filled in before being sent to the target system. Finally, you will set (or request) system settings according to a system logic that allows, disallows, and warns users in their work with Agile's business objects.

Attribute Properties

In completing the initial systems analysis, you should have identified the fields to be transferred from Agile to the target system. As a part of the requirements, state the field names, lengths, validations, and importance.

You can alter Agile attribute properties to match properties of the target fields, to reduce the possibility of data rejection. Using Agile Java Client, you can:

- Modify attribute labels to match those of the target system
- Reduce or increase default attribute lengths to match the target system counterpart (within the limits of the Agile field constraints)
- Define attribute formats

- Define list attributes that contain only valid entries for the field
- Force required attributes

Agile Java Client allows you to configure the Agile database with relative ease.

Modifying a Property

Target systems have different field lengths for their part numbers. If the part number in the target system is larger, there should not be a problem transferring the data from Agile to the target system. However, if the part number length in the target system is shorter, you can change the part number in Agile to match.

Note A field length change does not affect data already entered in the Agile database.

For details, see the *Agile PLM Administrator Guide*.

Validation List

You can add a validation list by opening an attribute that you can define as a List attribute and modify the select list.

For details, see the *Agile PLM Administrator Guide*.

Page Two - More Attributes

Additional attributes may be needed to accommodate data entry for fields like the unit of measure, source code (make or buy), class code, vendor, commodity code, and ABC code. While these are generally maintained from the manufacturing side, an initial entry may be required from Agile. Be careful to consider the following when setting up Page Two attributes:

- Will this field be modifiable after the initial release of the ECO?
- What will happen if this field and the field in the target system are not synchronized (which system has ultimate maintenance “say-so” for a given field)?
- Is this a required field for an addition? For a change?

Page Two attributes may be necessary for an initial release, but they might not be maintained in Agile because they are not under Engineering Control. There are two ways to handle this situation:

- Design the application adapter to write default values into fields that are not maintained in Agile, for an initial release only. Page Two attributes would then not be required for these types of fields, eliminating confusion and reconciliation problems later.
- Design the application adapter to allow the entries from Page Two attributes for the initial release of an item only. The application adapter would then have to ignore identified fields that would be sent with subsequent changes to items.

After careful consideration of what a form for additional information should consist of, the Page Two fields can be made visible and defined to meet the needs of the application adapter and customer. These attributes will be placed on the Page Two form for data entry. This form must be formatted and made visible before it is used.

For more detailed information, see the *Agile PLM Administrator Guide*.

Page Three

Agile allows subclassing, and provides for optional Page Three attributes at the subclass level. However, these fields are not mappable, and are not transferred by ChangeCAST. Any attributes that are to be transferred must occur at the class level.

For details, see the *Agile PLM Administrator Guide*.

Disallowing Modifications After Release

By disallowing modifications after release, you can keep an attribute from changing after release. Doing this requires modifications to roles and privileges for the Change Analyst and Incorporator.

For details, see the *Agile PLM Administrator Guide*.

SmartRules

You can configure the system logic to prevent certain critical errors from affecting the target system. For typical target system integrations with Agile, certain SmartRules are usually set, for example, “disallow” duplicate find numbers, “disallow” items released first. These SmartRules allow you to constrain your processes and help you provide valid data through the integration.

For details, see the *Agile PLM Administrator Guide*.

Data Formatting Limitations

The Agile Java Client includes the Characters property to place limitations on the kind of data that you can enter in a field. For details, see the *Agile PLM Administrator Guide*.

Transferring Files

Special data requirements can arise when transferring files, and matching data between Agile and the target system.

Examples

Example 1

If an attribute is a list field, there is no way to limit the number of characters for each item in that list. For example, the Unit of Measure list field may be set up in Agile Java Client with EA, BX, IN, CN, RL and DZ as valid entries to accommodate the target system valid entries.

Unaware of the ramifications, the system administrator takes the Agile Java Client class and determines that he can set up the entries as EACH, BOX, INCH, CAN, ROLL, and DOZEN instead. This will be valid in Agile, and, even though the ChangeCAST profile indicates that you need only two characters for the Unit of Measure field, the transfer file will be sent with the whole word (EACH instead of EA).

Example 2

The company has been using Agile and the target system for a while. However, the two systems were independent, and no consideration was given to the issues of matching field lengths and validations. You now need to integrate the two systems. A backlog of parts and BOMs pushes through Agile to update the target system on an as-needed basis.

Problems occur when an old part or BOM (pre-integration) needs change. The description of the part, when you enter in Agile is 100 characters long, but the target system allows only 30. ChangeCAST will not truncate after 30 characters (unless you modify the ChangeCAST profile with a special mapping condition for this field— although, truncating in ChangeCAST is not recommended). ChangeCAST will send the original 100 characters to be stored in the database.

Key fields need to have special attention in a situation like this. Simple truncation is likely to cause duplicate keys. If there is any significance to the part numbering, using any type of conditional mapping in ChangeCAST to reformat the part number will have to take this into consideration.

While you can manage this within ChangeCAST, it is probably better for the application adapter to:

- Take the (long) part number “as is” from Agile.
- Generate an error report for the part number.
- Provide a new numbering to the part, and release a new part on a new ECO from Agile.

Multitext and Multilist Fields

ChangeCAST has the ability to determine how the CR/LF will be represented in a text field in the transfer file. The formatting of data will replace CR/LF with the substitution character, and the whole field will be transmitted in one continuous string. Note that setting the CR/LF substitution character to none does not leave the CR/LF in the data, but rather just eliminates the position it took in the field.

Multilist fields are represented in the same way as reference designators. Each selection from the list is separated from the previous selection with a comma. In a horizontal orientation using double quotes, the field might appear like this:

…, “Project01,Project02,Project03”,…

Make sure that if a multilist is configured, the application adapter can interpret this output orientation properly.

Chapter 5

Creating a Profile

This chapter includes the following:

Before attempting to create a profile, you must have an understanding of what a profile is and how it is used to transfer data into an application adapter.

Understanding Profiles

The profile controls all aspects of the transfer of data from Agile to a target system, including file structure definition, transfer settings, scheduling information, data mapping, and file locations.

An understanding of the following terms will help in the development of the ChangeCAST profile. The table below defines some general terms, and describes the ChangeCAST components and the various ChangeCAST files involved.

Term	Definition
General Terms	
Attributes	A list of target system table-field elements that can be mapped; every profile contains a set of attributes. The terms *target system attribute* and *target system field* are used interchangeably in this manual.
ChangeCAST connect state	Whether ChangeCAST is automatically transferring change orders for enabled profiles; the possible states are Paused and Resumed, controlled by buttons in the main ChangeCAST window.
Field	An individual piece of information; an attribute or column of a table. The terms *target system attribute* and *target system field* are used interchangeably in this manual.
Mapping	The pairing of target system attributes with Agile fields, system nulls, system variables (also known as system flags), or the result of a conditional expression.
Profile	A collection of settings used by ChangeCAST to process and deliver the transfer file to the target system. Profiles define many aspects of data transfer from Agile to a single target system, including transfer file structure definition, transfer settings, scheduling information, data mapping, and directory/file locations.
Profile library	Up to 32 profiles together in a single file with extension .agx (for ChangeCAST 8.1 and later) or .agc (for previous releases).
Row	A grouping of fields or a record in a table that represents a data element.
Table	An organized collection of records.

Term	Definition
Transfer flag	An internal setting that tells ChangeCAST whether a change order has been successfully transferred.
ChangeCAST components	
ChangeCAST Service	A Windows 2000 service that launches the ChangeCAST application.
ChangeCAST.exe	A Windows NT or Windows 2000 application that: Allows users to create/edit/remove profiles (up to 32), which includes defining the transfer file format, the field mappings, and the transfer protocols. □ Sets up information required by the Agile ChangeCAST service. □ Allows viewing of the error logs. □ Allows user to start/stop transfers or run transfers manually. □ Allows user to reset transfer flags and resend change orders.
StartChangeCAST.exe	Application that starts the Agile ChangeCAST service.
ChangeCAST files	
Backup (.bak)	Backup of the Profile library.
Error log (ErrorLog.txt)	A log file that tracks ChangeCAST mapping, condition syntax, and other problems encountered during validations and audits.
Exported ChangeCAST attribute file (.aga)	An ASCII text file containing a list of target system tables and fields that are mappable for a single ChangeCAST profile.
Mapping file (.map)	An ASCII (text) file containing the mappings of Agile fields to target system fields.
Profile library (.agx or .agc)	A file that contains all aspects of one or more profiles. AGX files (for the current ChangeCAST) are text files. AGC files (the format for pre-8.1 releases of ChangeCAST) are binary files.
Temporary (.agl.tmp)	Temporary files created during processing.
Transfer file (.agl, by default)	A formatted ASCII file, output from ChangeCAST, which is used to transfer change order information from Agile to a host target system. There is at least one transfer file for every change order processed by ChangeCAST. Transfer files adhere to the settings defined in corresponding profiles.
Transfer log (TransferLog.txt)	A log file that tracks the status of ChangeCAST's connect state and transfer attempts.

Creating a Profile

Creating a profile can appear as an imposing task at first glance. The following steps guide you through the profile creation process.

To create a profile:

1. Create or load the attributes or fields from the target system that can be mapped for data transfer.
2. Map the target system fields to the Agile attributes.
3. Format the transfer file - set up the file structure, delimiters, data format, and BOM row settings.
4. Select the appropriate transfer settings.
5. Generate and archive reports for attributes, mappings, email, file, and transfer settings.

Suppose you, as the developer, create one or more profiles for a client, exit and stop ChangeCAST, and then copy the .agx file for delivery to the client. The client can then use **Profile | Import** to import the profiles.

Note The **Profile | Copy** command does not retain the options in the **Transfer** menu; these options must be defined for a copied profile.

Defining Attributes

To make the required associations between Agile and the target system, a list of all the transferable data fields is set up and maintained in ChangeCAST. This list, containing all or some of the fields or attributes for a target system, is then used to map Agile data fields to the target system data fields.

The attribute definition specifies the table name, table description, field name, field description, field length, and data type.

- The table name refers to the table or dataset name that you target to receive information. The table description is the table's user-friendly description.
- The field name and field description refer to the actual attribute or field to receive information.
- The field length is the maximum field length allowed for this attribute or field on the target system.
- The data type is the type of data allowed: numeric, date, or text.

You can enter these attributes and maintain them in ChangeCAST. A good practice in defining the attributes is to list all fields for the affected target system tables. Doing this provides flexibility if it is determined that an additional field will be maintained in Agile at some later point in the implementation.

Note You should map only those attributes that the adapter supports. Doing otherwise can cause inconsistent behavior or may cause the adapter to fail and stop processing.

Select the profile, and then use the **Attributes** commands in the **Mapping** menu (**Add Attributes**, **Import**

Attributes, **Edit Attributes**) to create the attribute definition. These attributes will then be available for mapping. For specific steps, see “Defining Attributes and Field Mappings,” in the *Agile ChangeCAST User Guide.*

Equation 1: -1: Attribute Table window

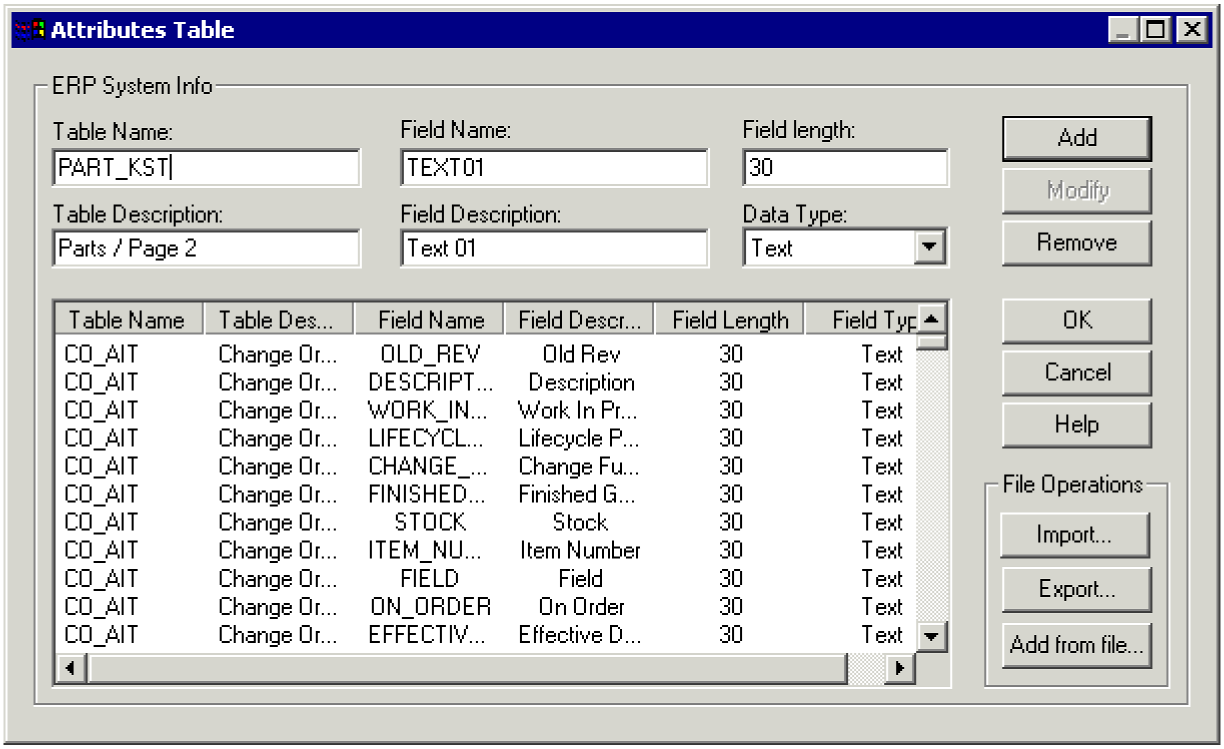


Table Name: PART_KST

Table Description: Parts / Page 2

Field Name: TEXT01

Field Description: Text 01

Field length: 30

Data Type: Text

Table Name	Table Des...	Field Name	Field Descr...	Field Length	Field Typ
CO_AIT	Change Or...	OLD_REV	Old Rev	30	Text
CO_AIT	Change Or...	DESCRIPT...	Description	30	Text
CO_AIT	Change Or...	WORK_IN...	Work In Pr...	30	Text
CO_AIT	Change Or...	LIFECYCL...	Lifecycle P...	30	Text
CO_AIT	Change Or...	CHANGE_...	Change Fu...	30	Text
CO_AIT	Change Or...	FINISHED...	Finished G...	30	Text
CO_AIT	Change Or...	STOCK	Stock	30	Text
CO_AIT	Change Or...	ITEM_NU...	Item Number	30	Text
CO_AIT	Change Or...	FIELD	Field	30	Text
CO_AIT	Change Or...	ON_ORDER	On Order	30	Text
CO_AIT	Change Or...	EFFECTIV...	Effective D...	30	Text

Mapping Data from Agile to the Target System

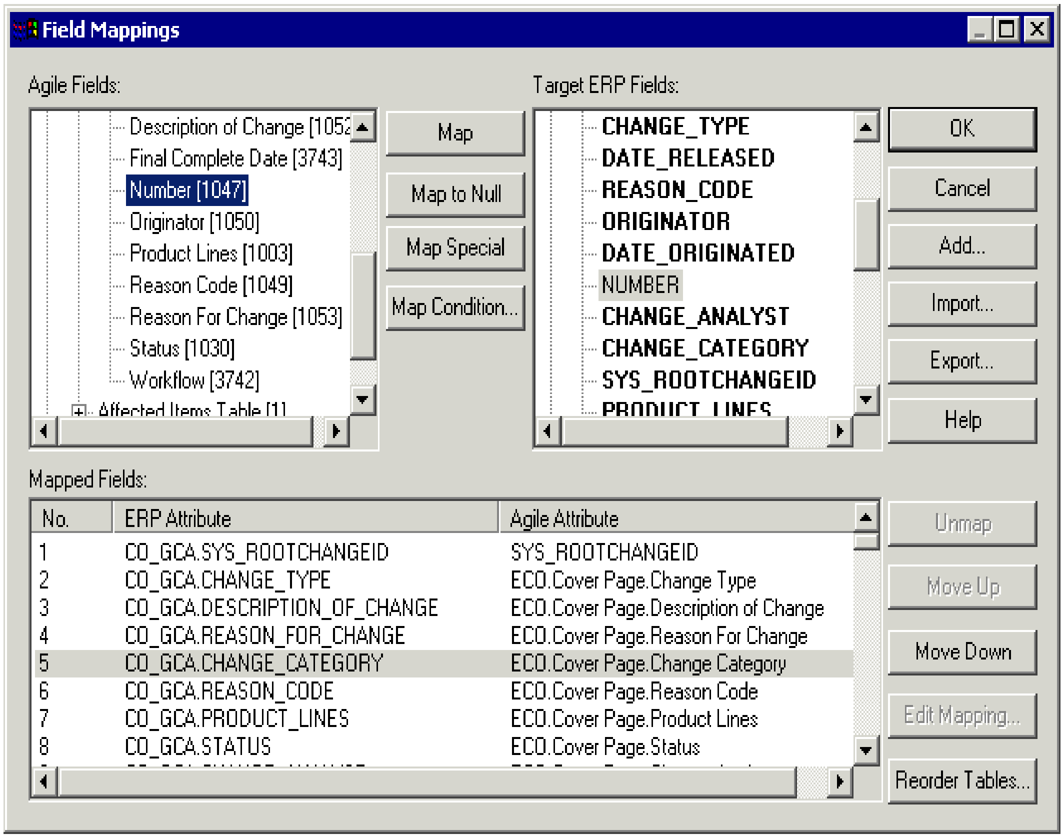
Typically, mapping refers to the field-to-field relating of data. However, there may be a situation that you need to deal with, based on information that the user will supply.

For example, a part may need to be available to manufacturing and sales. Because another table maintains sales parts, you could add and map a Page Two attribute called “Add to Sales?” Based on user response, the application adapter would determine that you need to add this part to the sales table as well as the manufacturing table. While this attribute may not have a target system database equivalent, it will have programmatic merit within the application adapter itself.

- The field names assigned in Agile Java Client and the attributes display side by side in the Field Mappings dialog box of ChangeCAST.

Field Mappings dialog box

Equation 2: -2: Field Mappings dialog box



No.	ERP Attribute	Agile Attribute
1	CO_GCA.SYS_ROOTCHANGEID	SYS_ROOTCHANGEID
2	CO_GCA.CHANGE_TYPE	ECO.Cover Page.Change Type
3	CO_GCA.DESCRIPTION_OF_CHANGE	ECO.Cover Page.Description of Change
4	CO_GCA.REASON_FOR_CHANGE	ECO.Cover Page.Reason For Change
5	CO_GCA.CHANGE_CATEGORY	ECO.Cover Page.Change Category
6	CO_GCA.REASON_CODE	ECO.Cover Page.Reason Code
7	CO_GCA.PRODUCT_LINES	ECO.Cover Page.Product Lines
8	CO_GCA.STATUS	ECO.Cover Page.Status

Note Ensure that you re-start ChangeCAST after you make changes to Agile Java Client, or the changes to fields may not reflect in ChangeCAST.

- Remember that the order of tables and fields is important. The application adapter or processor usually requires that tables and fields be in a specific order.
- You can import and export mappings. Field mapping files are text files with a .MAP filename extension. They are created when a user exports mappings from ChangeCAST profile.

For more information about mapping fields, see “Defining Attributes and Field Mappings,” in the *Agile ChangeCAST User Guide.*

Mapping to a specific target system field

To map an Agile attribute to a specific target system field:

1. Expand the desired table node by clicking the plus sign next to the table name.
2. Click the Agile field.
3. Click the target system field.
4. Click **MAP**. Notice that an entry is made in the **Mapped Fields** section of the window describing

the target field with an expression. The expression points to the specific Agile attribute and includes the class, table name, attribute name, and attribute ID. This attribute ID enables backward compatibility with previous versions of ChangeCAST.

Mapping Agile AML Fields

The following considerations apply if you are mapping AML fields:

- Ensure that fields are set to **Visible** on Agile Java Client **Parts Class | Manufacturers | Attributes** tab.
- Map all Agile manufacturer fields to a single target system table, separate from other BOM item data.
- You cannot map multilist fields of manufacturers and manufacturer parts.

In addition, fields on the optional **Page Three** tab are not mappable. Ensure that all fields you need to transfer are on **Page Two** or in other tab, where they are mappable. See **Appendix C** for details on manufacturer parts.

Mapping to Null

When you click a target system field and **Map to Null** it maps null to the field specified. Why map this field to null?

To provide room for growth and flexibility, it is always a good idea to identify (in the attribute table) and map (through the field mappings) all the fields. If a new field mapping is required later, it can be modified easily. In addition, a field mapped to null can sometimes be used as a flag to the application adapter that a default target system setting should be used.

Mapping a field to null will return the character string for Nulls specified in the **Data Format** tab of the File Layout Settings dialog box (**Format | Data Format**) to the application adapter, for processing as a "blank" or "Null" value.

Mapping to System Variables

Use **Map Special** to map target system fields to system variables, or system flags, such as SYS_ACD.

Using ChangeCAST Macro Language

Use **Map Condition** to use the ChangeCAST Macro Language to populate the target system field with conditional data. That data can be a default value, a substring of a field, a concatenation of several fields, an array-oriented table, and much more.

To open the Map Condition dialog box to revise a mapping, click a field in the **Mapped Fields** list, and click the **Edit Mapping** button. For more information, see Appendix A - ChangeCAST Macro Language. For information about using the Map Condition dialog box, see Chapter 3 of the *Agile ChangeCAST User Guide*.

Mapping Defaults

When sending defaults to the application adapter, consider how the field should be dealt with during an initial release as well as for subsequent changes. A macro can be used to send the correct field entry depending on whether this is an initial release or a change record.

Mapping BOM Fields

When you set up field mapping for the BOM table, there is only one reference to a part number ([Parts.BOM Table.Item Number:1011]). This refers to the “child” part number. To make an association to the “parent” part number, map Parts.General Item Attributes.Number:1001 to the target system’s equivalent attribute.

Mapping Different Information to Different Tables

Follow these guidelines to mapping different types of information:

- AML and BOM information should not be mapped to the same table.
- Pending change information should not be in the same table with AML or BOM information.

Formatting the Transfer File

Once you configure Agile Java Client, load attributes, and map fields, determine the format of the transfer file. The appropriate format depends on how you provide data to ensure data integrity when you populate the target system’s tables.

You can configure the following aspects of formatting the transfer file from within ChangeCAST:

- File Structure on page 29
- Delimiter on page 30
- Data Format on page 30
- BOM/AML Table Settings on page 32

A command in the Format menu and a corresponding tab in the File Layout Settings dialog box represent each of these. Once you establish this, the settings correspond to expectations within the parsing logic of the adapter. This applies to file structure, delimiter, data format, and BOM/AML table settings.

File Structure

Use the **File Structure** tab of the File Layout Settings dialog box to select horizontal or vertical orientations of the transfer file. You can also specify header text in the **Prefix**, **Layout Section**, and **Data** fields.

You can use the check boxes in the Output section to select or clear the following settings:

- **Table headers** – Checking this box indicates that you want table headers. Un-check the box to remove headers.
- **Field Names (Layout)** – Checking this box indicates the field names you want in the **Layout Section** field.
- **Blank Lines Between Tables** – If you send all data in a single file - and check this box, it indicates that you want to insert a blank line between data types.

- **Empty Tables if no Rows** – If no data is sent for a specific target system table - and check this box, it indicates that you want ChangeCAST to supply an empty table (by generating just the header text).
- **ECO Number and Line Count** – Checking this box indicates that you want ChangeCAST to generate a footer with the ECO number for the file as well as the number of lines in the transfer file.

Notice that the sample file shown in the File Layout Settings dialog box changes dynamically as you select options, representing what the actual transfer file will look like.

Delimiter

You can specify the following types of delimiter information:

- **Type of Data** – Determines whether you need to format as delimited or fixed width. Use **Delimited** option with vertical orientations.
- **Text Delimiters** – List of common text delimiters. This option is configurable only for the horizontal orientation. The delimiters provided are None (no delimiter), " (double quote), and ' (single quote).
- **Field Delimiters** – Check boxes provide the common field delimiters: comma, space, tab, semicolon, and other (a fill-in-the-blank field). In vertical orientation, the field delimiters control the separation of the headers.
- **Line Delimiters** – The supported options are: **CR/LF (ANSI - Windows)**, **LF (ASCII - Unix)**, and **CR (Macintosh)**.
- **Data Type** – Check the box if the target system requires delimiters with Numeric data.

You can view changes made in the **Delimiter** tab in the sample file on the **File Structure** tab.

Note When designing the application adapter, ensure that any character used as a delimiter can also be handled as part of an incoming text field. For example, if you use the double quote as a text delimiter, make sure that the application adapter understands the structure.

- The first character of the record is one double quote.
  - Fields are delimited with a combination of double quote-comma-double quote.
  - The record terminates with one double quote and the line delimiter. For example: **"<field text 1>","<field text 2>","<field text 3>"<cr/lf>**
- If you use a line or field delimiter in the description of a field, it should not confuse the application adapter.

Data Format

Settings in the **Data Format** tab determine the representation and formatting of specific data fields.

Reference Designators

Reference designators are stored in ChangeCAST individually. In generating a transfer file, ChangeCAST groups the reference designators together for the identified BOM row and generates output based on the selection of reference designators made in the **Data Format** tab of the File Layout Settings dialog box.

Be sure you understand how reference designators are handled in the Target System, to select the correct format:

- **Individual** and **Compressed** – Individual format separates each reference designator with a comma (R1, R2, R3, R5, …). Compressed format indicates a range using a hyphen (R1-R3, R5, …).
- **Normal** – Lists reference designators as they appear in Java Client.
- **One Designator per line** – This option is available for both horizontal and vertical formats and acts in conjunction with the **Individual** and **Compressed** options to define the reference designator format. If **Horizontal** is selected on the **File Structure** tab, a full row is output with a single group of reference designators.
- **Wrap Ref. Designators** – Lists reference designators and ranges one after the other on a line, then wraps (that is, a second data entry is created into which the reference designators field continues).

Horizontal or Vertical format	One reference designator per line	Multiple reference designators per line
Individual	R1 R2 R3 R5	R1,R2,R3,R5
Compressed	R1-R3 R5	R1-R3,R5

If you want the Data Formats to apply to reference designators, map the reference designators directly to the appropriate field in the user data. If you use a conditional mapping, the Ref Des field treats it as a string and ignores the Data Format settings.

Due to field length limitations on several target systems, commas do not follow spaces in the groupings that allow multiple reference designators per line.

Note Compressed format refers to the first number of the range to set the number of digits in each number in the range. For example, if the first reference designator is R1, then the maximum value it can take in the range is R9. If the first reference designator is R01, then the maximum value it can take in the range is R99. Therefore, with this option set to Compressed, B1-B100 becomes B1-B9, B10-B99, B100.

The Sample field in the dialog box displays a sample of the reference designator format.

Time and Date Format

The available date formats are the following:

M/d/yyyy	MM/dd/yyyy	d/M/yyyy	dd/MM/yyyy

d-MMM-yyyy	DD-MMM-yyyy

(M = 1- or 2-digit month, MM = 2-digit month, MMM = name of month, d = 1- or 2-digit day, dd = 2-digit day, yyyy = 4 digit year)

Note To handle Year 2000 (Y2K) requirements, Agile uses only a 4-digit year format.

Because of this, on a Windows NT server, you must choose one of the following as the short date format in Control Panel **|** Regional Settings **|** Date: MM/dd/yyyy or M/d/yyyy.

The available Time formats are the following:

h:mm:ss tt	hh:mm:ss tt	H:mm:ss
HH:mm:ss	HH.mm	

(h = 1- or 2-digit hour as appropriate, hh = 2-digit hour, H = 1- or 2-digit military hour, HH = 2-digit military hour, mm = 2-digit minutes, ss = 2- digit seconds, tt = AM/PM)

Conversions

Use the fields in this section to alter Agile text fields so they can be properly handled by the application adapter. The following selections are available:

- Convert SQL Nulls in text to either ## or spaces. The default is ##. When you check the **Use Text Delimiters** box, the text delimiter selected in the **Delimiter** tab will be used to encase the character selected for SQL Nulls.
- Convert CR/LF used in text fields (or in a single column) to one of the following: space, ASCII 254, ^ (caret), + (plus), or nothing (null). The application adapter must convert back to the original.
- Convert instances of the text delimiter that occur within a field to either ASCII 135, \" (backslash *text delimiter*), or "" (*text delimiter text delimiter*). To use this setting, you must set the orientation to horizontal and the text delimiter to something other than None. The application adapter must convert back to the original.

BOM/AML Table Settings

Use the **BOM/AML Table Settings** tab to configure BOM row output. You can configure the previous revision and initial release; the table at the right lists the target system tables. (This table is read-only. ChangeCAST creates the target system–Agile table designations automatically.)

Full vs. Delta

- The **Full** option will output to the transfer file all rows (even the unmodified ones) for the affected single-level assembly.
- The **Delta** option formats output only for items that have changed on an affected assembly. Unmodified components are not included in the transfer file.

These options apply only when a previous revision exists, since, on an initial release, all items are treated as additions.

Initial Release Output of Effective Date

Check **Effective Date** if you need to send the effective date assigned to the affected par.

Check **As Text**, rather than send the effective date assigned. ChangeCAST should send a string constant, such as 00/00/0000.

These options may apply only to an initial release of an assembly and its components.

Target System – Agile Tables Designation

In the **Target System – Agile Tables Designation** table, the left column lists the target system tables extracted from the attribute file for the application adapter, as specified in the selected profile. The right column indicates the Agile table to which the target system table is linked. Linked target system tables will contain BOM row information (assembly, component, find number, quantity, and reference designators).

Note If you have a target system table with BOM information, ChangeCAST sets it as a BOM table even if this table was not marked as a BOM table by pre-8.1 releases of ChangeCAST. This means that ChangeCAST will transfer different information. If you want the information transferred to be the same as with previous releases, choose **Mappings | Edit Mappings** and replace the target with 0 (zero).

The following table illustrates how a target system table designated as a BOM table will represent BOM data in the transfer file.

Behavior	Description	Sys_acd
Add	New record with effective date only; obsolete date sent as a null value.	a
Change	Old records with obsolete date only; effective date sent as a null value. New record with effective date only; obsolete date sent as a null value.	c a
Delete	Old records with obsolete date only; effective date sent as a null value.	d
Unmodified Row	One record with effective and obsolete date sent as null values if using full bom orientation.	u

You can map the SYS_ACD flag shown in the right column to a program attribute for use with the BOM table. It will return an A, C, D, or U (for Add, Change, Delete, or Unmodified) depending on the BOM row settings. A full BOM may generate output with A, C, D, and U, while a Delta BOM will generate only output with A, C, and D. Note that changes to a single BOM row will generate two records, one flagged as a change (C), and the other flagged as an add (A).

You can map the same flags used with SYS_AMLACD to a program attribute for use with the Manufacturer table in the same way.

Sample Profiles

Several profiles are delivered as samples that include only the file format settings required for generating the output described here. The filename is sample_ChangeCAST.agx, and by default, it is installed in the directory Agile\ChangeCAST\Profiles.

To get a full report of all the file format settings, you can run the Profile Settings report. See, “Defining Attributes and Field Mappings,” in the *Agile ChangeCAST User Guide* for details.

The Generic Full Profile

The Generic Full Profile orients the file horizontally and generates data for all rows (changed and unchanged) of an affected simple assembly. Each field is separated only by a comma, and all data is double quote–encased (nulls ## are not double quote–encased). The following is sample output from the Generic Full Profile:

```
# <table #1 name> Data
"<field #1 data>","<field #2 data>","<field #3 data>",##,
"<field #5 data>"
"<field #1 data>","<field #2 data>","<field #3 data>",
"<field #4 data>",##

# <table #2 name> Data
"<field #1 data>","<field #2 data>","<field #3 data>"
"<field #1 data>","<field #2 data>","<field #3 data>"
```

Note that in the first record in table #1 there is no data for field #4, which is represented by ##. In the second record in table #1, there is no data for field #5, and it is also represented by ##.

The Generic Delta Profile

The Generic Delta Profile generates a vertically oriented file. The output includes only data for the affected BOM rows on an assembly or any affected component change. Each field is represented on a separate line delimited by a CR+LF (ASCII 32). Nulls are sent as ##. The following is sample output:

```
# <table #1 name> Layout
<table #1 name>,<field #1 name>,<field #1 length>
<table #1 name>,<field #2 name>,<field #2 length>
<table #1 name>,<field #3 name>,<field #3 length>
<table #1 name>,<field #4 name>,<field #4 length>
<table #1 name>,<field #5 name>,<field #5 length>
# <table #1 name> Data
<record #1, field #1 data>
<record #1, field #2 data>
<record #1, field #3 data>
##<record #1, field #4 NO data>
<record #1, field #5 data>
<record #2, field #1 data>
<record #2, field #2 data>
<record #2, field #3 data>
<record #2, field #4 data>
##<record #2, field #5 NO data>
<blank line between tables>
# <table #2 name> Layout
<table #2 name>,<field #1 name>,<field #1 length>
<table #2 name>,<field #2 name>,<field #2 length>
```

```
<table #2 name>,<field #3 name>,<field #3 length>
# <table #2 name> Data
<record #1, field #1 data>
<record #1, field #2 data>
<record #1, field #3 data>
<record #2, field #1 data>
<record #2, field #2 data>
<record #2, field #3 data>
```

Data Representation

Agile uses the following model for BOM data output, to accommodate the multitude of target systems Agile is interfacing with today. The following example used the mapping: Assembly, Component, Find Number, Quantity, Effective Date, Obsolete Date, Reference Designator, BOM, Notes.

Adds

Adding a BOM row's component parts generates a single record in the output file. The record has all the user-mapped fields as well as any mapped defaults. Assembly BOM row records have an effective date equal to that of the assembly on the **Affected** Items tab, and an obsolete date of null (##), unless the **Output Effective Date** option is used. Note how the SYS_ACS flag reflects the new BOM row by using an "A" (representing "add"). In the example, the fields are as follows:

```
"Assembly Number","Component Number","Item Number","Quantity
Per","Effective Date","Obsolete Date", "Reference
Designator","Notes,"SYS_ACD
```

Note the position of the closing parenthesis (").

```
# BOM Data
"GF-200-00","GF-200-01",1,"6","12/11/1997",##,##,##,A <record to
make new part effective>
"GF-200-00","GF-200-02",2,"1","12/11/1997",##,##,##,A <record to
make new part effective>
```

Changes

Changes to an assembly's BOM rows generate two records in the output file. The first record contains the information from the prior BOM row and includes the obsolete date from the assembly entry in the **Affected Items** tab with an effective date of null (##). The second record contains the new BOM row information and includes the effective date from the assembly entry in the **Affected Items** tab; the obsolete date is null (##). This example represents the Full BOM. Note how the SYS_ACD flag reflects the changes by using an "C" (representing "change").

```
# BOM Data
"GF-100-00","GF-200-00",1,"1",##,##,##,##,U <unchanged row>
"GF-100-00","GF-300-00",2,"1",##,##,##,##,U <unchanged row>
"GF-100-00","GF-500-00",4,"2",##,##,##,##,U <unchanged row>
"GF-100-00","GF-501-00",5,"1",##,##,##,##,U <unchanged row>
"GF-100-00","GF-600-00",6,"1",##,"10/3/1997",##,##,C <record to
obsolete the row>
"GF-100-00","GF-600-01",6,"1","10/3/1997",##,##,##,A <record to make
new part effective, changing the part number>
"GF-100-00","GF-700-00",7,"4",##,"10/3/1997",##,##,C <record to
obsolete the row>
"GF-100-00","GF-700-00",7,"5","10/3/1997",##,##,##,A <record to make
new part effective, changing quantity>
```

Deletes

Deletion of a component part from an assembly generates a single record in the output file. Agile re-sends all mapped and default fields regardless of whether they have been changed or not. Be careful when writing an interface to make sure that no part is deleted, since it may be used on other BOMs or in another part of an integrated database. The deletion record contains the current BOM row information as well as the obsolete date from the assembly record on the **Affected Items** tab. The effective date is null (##). This example represents the Full BOM. Note how the SYS_ACD flag reflects the deletion by using a "D" (representing "delete")

```
# BOM Data
"GF-1000-00","GF-2000-00",1,"1",##,##,##,##,U <unchanged row>
"GF-1000-00","GF-3000-00",2,"1",##,##,##,##,U <unchanged row>
"GF-1000-00","GF-4000-00",3,"1",##,"9/3/1997",##,##,D <record to
obsolete the row>
"GF-1000-00","GF-5000-00",4,"2",##,##,##,##,U <unchanged row>
"GF-1000-00","GF-5001-00",5,"1",##,##,##,##,U <unchanged row>
"GF-1000-00","GF-6000-00",6,"1",##,##,##,##,U <unchanged row>
"GF-1000-00","GF-7000-00",7,"4",##,##,##,##,U <unchanged row>
```

Unmodified Records

For a full BOM, unmodified rows for a BOM are represented in a single record with all the current information for the row, but the effective and obsolete dates are null (##). Note in the above examples how several records are formatted and do not have an effective or obsolete date; these are unmodified BOM rows for this released change order. When **Delta BOM** is selected, these rows are not transmitted.

Creating Transfer Settings

Setting up the parameters that govern transfer of the data is done by configuring the tabs of the **Transfer Settings** dialog box. See "Formatting and Transferring Agile Data Files," in the *Agile ChangeCAST User Guide.*

The following options are available from the **Transfer** menu; each one represents a tab in the Transfer Settings dialog box:

- Protocol Tab on page 36
- Notification Tab on page 37
- Schedule Tab on page 38
- Files Tab on page 40
- Application Adapter Tab on page 41

Protocol Tab

Choose **Transfer | Protocol**, or click **Transfer Options** in the Add Profile dialog box, to open the **Protocol** tab.

To gather information to enter in the FTP Protocol dialog box for a successful FTP connection:

Capitalize as shown in this example, which uses an H-P system.

1. Click **Start**.
2. Choose **Programs | Accessories | Command Prompt**.
3. At the DOS prompt, type **ftp** and press Enter.
4. At the ftp> prompt, type **open** *dnsname* (or **open** *IP address*) and press Enter**.** *dnsname* or *IP address* corresponds to the FTP hostname in the dialog box.
5. Enter user and password when prompted.
6. After receiving a "230 User logged on" message, type **pwd** and press Enter. The current directory is displayed. Make note of the exact directory notation as it appears in the first 257 message. This is used for the FTP target path.
7. Using a sample file (in the local directory), type **put** *filename* and press Enter. You will see "Transfer Complete" if the FTP has been successful.
8. Type **dir** and press Enter. The file FTP'd in the last step should be visible.
9. Type **disconnect** *dnsname* (or **disconnect** *IP address*) and press Enter.
10. To exit, type **bye** and press Enter.
11. Use the **Verify FTP Connection** button for confirmation after entering the information in the **Protocol** tab.

Be sure the target system does not truncate or alter files transmitted by FTP.

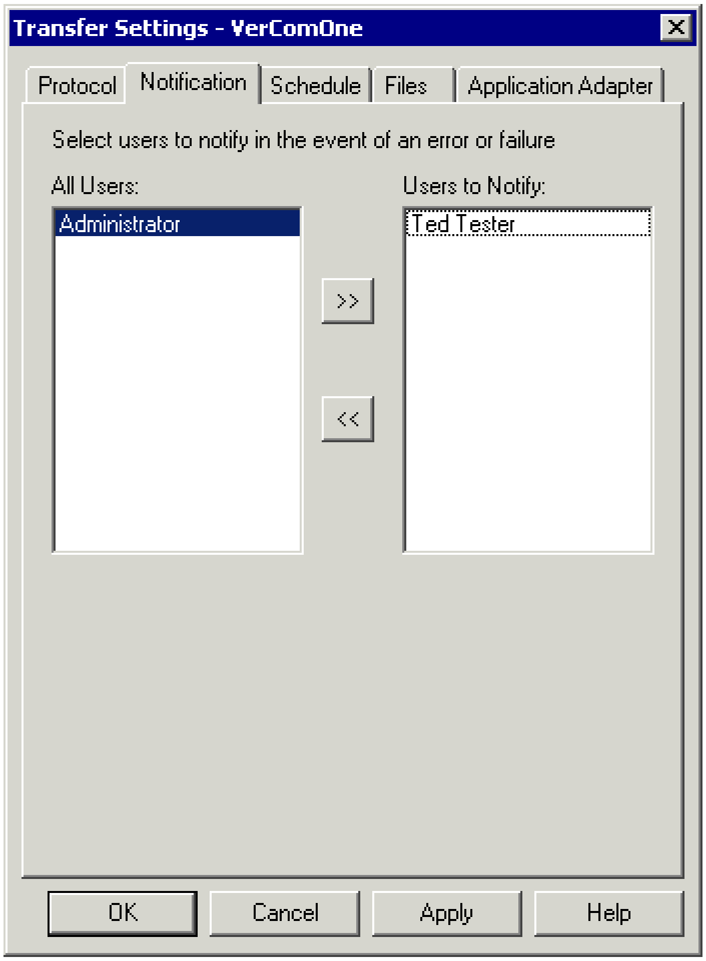
See, "Formatting and Transferring Agile Data Files," in the *Agile ChangeCAST User Guide.*

Notification Tab

Use these settings to select users you need to notify in the event of an error or failure in the ChangeCAST FTP process.

Note Agile Java Client establishes and maintains Email addresses and paths.

Equation 3: -9: Notification tab in the Transfer Settings dialog box



Select users to notify in the event of an error or failure

All Users: Administrator

Users to Notify: Ted Tester

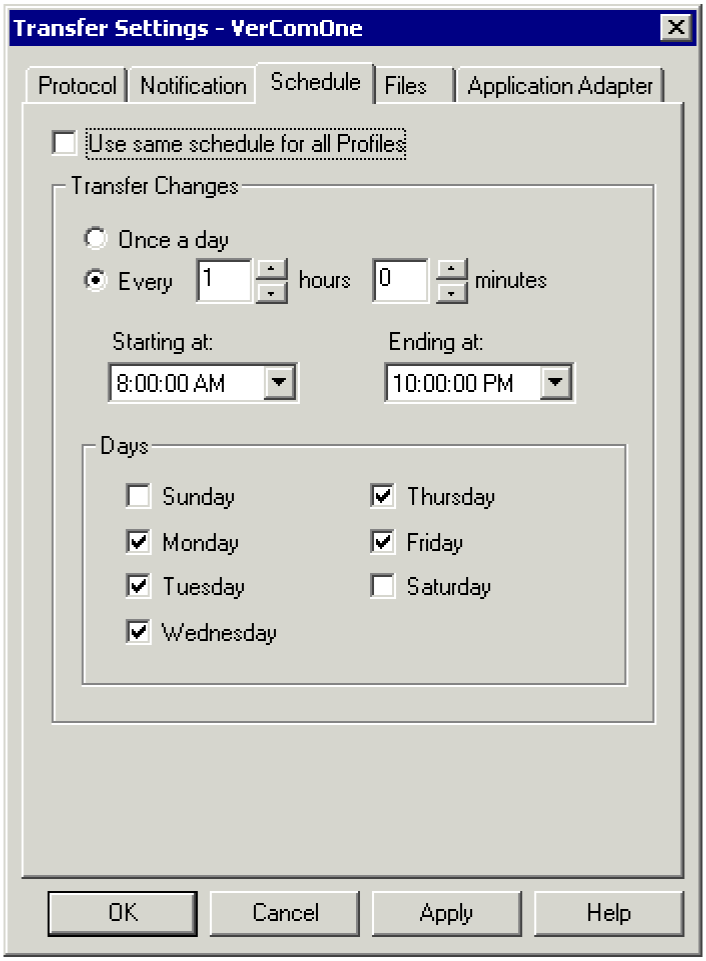
If there are changes to the Users to Notify list, ChangeCAST will not detect them until you re-start it. See “Formatting and Transferring Agile Data Fil*es,” in* the *Agile ChangeCAST User Guide* for details.

Schedule Tab

Configure the schedule to specify when to poll Agile database for released change orders and transfer them.

The Agile ChangeCAST service uses the start/end times to determine whether to process change orders. If the ChangeCAST state is Resume (running) and the current time is between the end and start times (down time), it will not process any change orders until the start time is reached. If the state is Paused, then the Agile ChangeCAST service stops regardless of the schedule until you resume it.

Equation 4: -10: Schedule tab in the Transfer Settings dialog box



☐ Use same schedule for all Profiles

Transfer Changes

○ Once a day

● Every 1 hours 0 minutes

Starting at: 8:00:00 AM

Ending at: 10:00:00 PM

Days

☐ Sunday

☑ Monday

☑ Tuesday

☑ Wednesday

☑ Thursday

☑ Friday

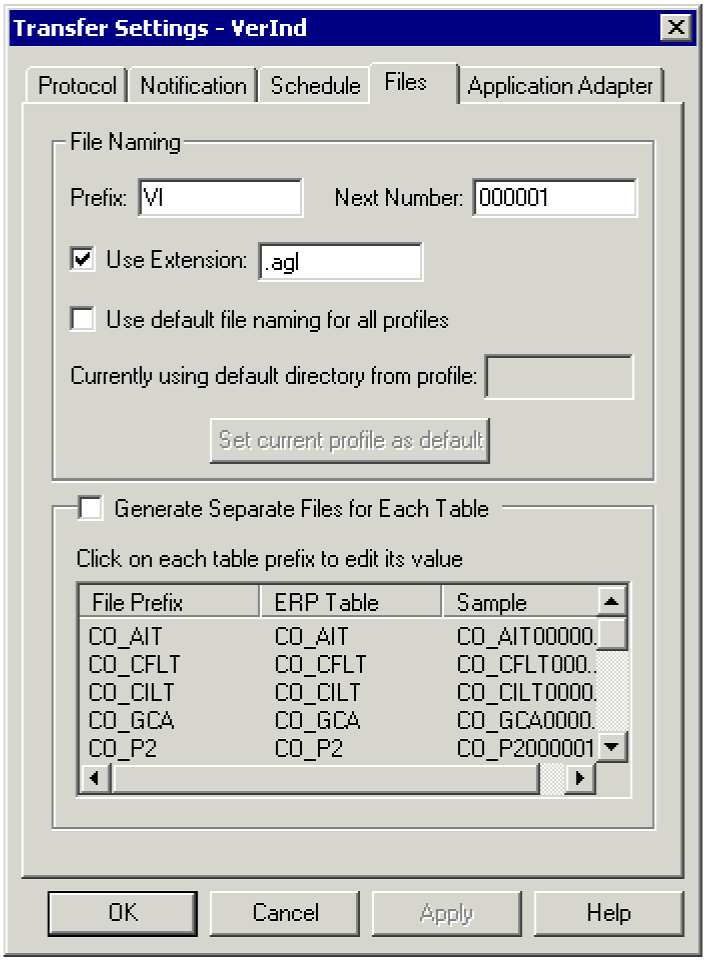
☐ Saturday

Also, see “Formatting and Transferring Agile Data Files,” in the *Agile ChangeCAST User Guide*.

Files Tab

This section contains file options conventions for transfer files.

Equation 5: -11: Files tab in the Transfer Settings dialog box



File Naming

Prefix: VI Next Number: 000001

☑ Use Extension: .agl

□ Use default file naming for all profiles

Currently using default directory from profile:

□ Generate Separate Files for Each Table

Click on each table prefix to edit its value

File Prefix	ERP Table	Sample
CO_AIT	CO_AIT	CO_AIT00000.
CO_CFLT	CO_CFLT	CO_CFLT000..
CO_CILT	CO_CILT	CO_CILT0000.
CO_GCA	CO_GCA	CO_GCA0000.
CO_P2	CO_P2	CO_P2000001

Naming the file

Options in this section are as follows:

- **Use default file naming for all profiles —** Indicates whether to use one file naming sequence for all profiles. When checked, if 3 profiles process 10 change orders, the first uses numbers 1-10, the second uses 11-20, and the third uses 21-30. When not checked, all profiles would use 1-10. When this box is checked, the bottom section of this tab is disabled.
- **Prefix** – The prefix used for the transfer file.
- **Next Number** – Indicates the number to be used when generating the next file.
- **Use Extension** – Indicates whether a filename will have an extension. If checked, the user enters an extension.

- **Currently using default file naming from profile** – The name of the profile that is currently using these default file naming conventions. Available only if **Use default file naming for all profiles** is checked.
- **UseSet currentl profile as default** – Sets the file naming conventions of the currently selected profile as the default for all profiles. Available only if **Use default file naming for all profiles** is checked.

Applying the File Prefix

Options in this section are as follows:

- **Target System Table** – The name of the target system table, which may be generated separately. This list is populated with the table names from the Attribute file.
- **Next File Name** – The next filename in the appropriate sequence for this table. This combines the prefix with the numbering sequence and any extension (as specified in the File Naming section described above). If **Generate Separate Files for Each Table** is not checked, all the tables will reflect the file-naming scheme as entered in the File Naming section. If **Generate Separate Files for Each Table** is checked, the initial defaults for the prefixes will be the same as the File Prefix above.

Generating Separate Files

Options in this section of the **Files** tab are available only if you do *not* check the **Use default file naming for all profiles** box.

- **Generate Separate Files for Each Table** – Check this box to generate separate files for each target system table. If the target system table is empty, no file is generated.

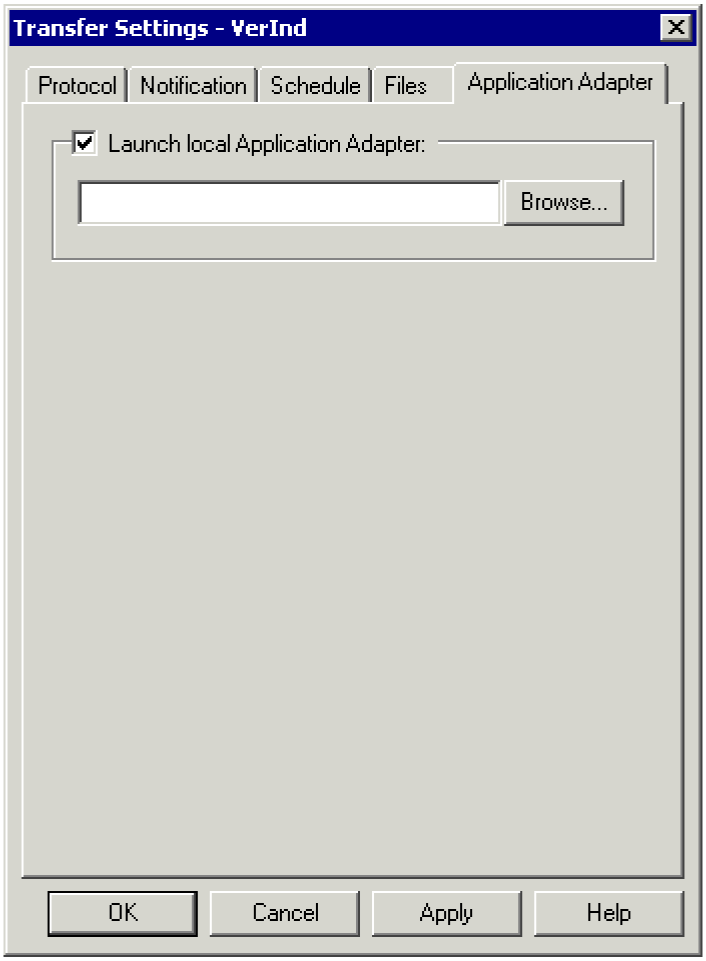
  If Generate Separate Files for Each Table is not checked, all the tables will reflect the file naming scheme entered in the File Naming section. If it is checked, the initial defaults for the prefixes will be the same as the **Prefix** in the top section of the dialog box.
- **File Prefix** – The prefix for the target system table is selected in the table. Click a value in this column to edit a table's prefix. Click **Apply** to apply the prefix to the table.

Application Adapter Tab

The **Application Adapter** tab contains the option for launching your application adapter, provided it is local to the ChangeCAST server.

Note You can use this option to run any local process such as an application adapter or system maintenance utility. Because it is local, this option is not available if files are FTP'd to the host system. In addition, you cannot choose application adapters installed on mapped or remote drives.

Equation 6: -12: Application Adapter tab in the Transfer Settings dialog box



☑ Launch local Application Adapter:

- **Launch Local Application Adapter** – Check this box and fill in the appropriate, fully qualified executable or batch file to launch after ChangeCAST successfully generates and delivers the transfer files. The program launched can be either the application adapter or other post-processor, provided it resides on the local machine (Agile server). If this feature is used to launch the application adapter, the application adapter does not require its own scheduling module.

Note If a post-processor is required, it must be able to deal with multiple files, processing them in the correct order. This ensures easier troubleshooting and provides stop-gaps if change order dependencies are more frequent than the processing schedule.

Generating Reports

ChangeCAST provides several reports. You can generate them in a simple text format with a .txt extension and view them with Notepad™. You can generate these reports for the current profile or for all profiles within the profile library.

ChangeCAST also generates two logs, which are available to view from the main menu, the error log (called ErrorLog.txt) and the transfer log (TransferLog.txt).

Reports

You can generate following reports:

- Attributes
- Mappings
- Settings
- Transfer Options

It is a good idea to create a hard copy of these reports after configuring ChangeCAST to generate the desired output. The reports, together with a software backup, can assist in recovery from problems.

The Agile Tables report lists the Agile attributes that are visible and therefore able to be mapped. It lists the subclass, page (or tab), the name of the attribute, and the ID number of visible attributes.

The following is an excerpt from the beginning of an Agile Tables report:

Agile Table

- **SubClass: Change Orders**
  - **Page: General Change Attributes [0]**
    - "Product Lines" [1003]
    - "Status" [1030]
    - "Reason For Change" [1053]
    - "Description of Change" [1052]
    - "Change Category" [1060]
    - "Change Analyst" [1099]
    - "Number" [1047]
    - "Date Originated" [1061]
    - "Originator" [1050]
    - "Reason Code" [1049]
    - "Date Released" [1051]
    - "Change Type" [1069]
  - **Page: Page Two [6]**
    - "Text01" [2035]
    - "List01" [2048]
    - "Date01" [2030]
  - **Page: Affected Items Table [1]**
    - "Stock" [1086]
    - "New Rev" [1056]
    - "Obsolete Date" [1078]
    - "Effective Date" [1079]
    - "On Order" [1085]
    - "Field" [1089]
    - "Item Number" [1054]
    - "Old Rev" [1055]
    - "Finished Goods" [1088]
    - "Change Function" [1058]
    - "Lifecycle Phase" [1057]
    - "Work In Progress" [1087]
    - "Item Description" [1059]
  - **Page: Change Image List Table [2]**
    - "Image Description" [1042]
    - "Checkout Date" [1147]
    - "Checkout Folder" [1148]
    - "File List" [1103]
    - "Checkout User" [1146]

Logs

The error log is a transaction log and contains results from audits of the profile, requests to change the transfer flag, and report requests.

The transfer log shows the change order to be transferred, the file number assigned, changes to the state (Resume and Pause), runtime errors, and the location of the file. The logs are text format files, and you can view them using a Notepad.

Chapter 6

System Testing

This chapter includes the following:

The development process must include end-to-end testing of the application adapter. Whenever possible, someone other than the primary application adapter programmer should perform the system tests.

As part of the test process, review target system reports on the data transferred from Agile, to ensure that existing business reporting processes function with the application adapter in place.

Setting Up the Test Environment

The test environment should include the appropriate version of Agile software, the target system software, the application adapter, and a post-processor or data processor, if required. Ensure to record all revisions used in testing.

Setting Up the Agile Server

Testing is best performed on an Agile server that runs only the software you need to verify. This ensures that Windows 2000 version and patches, registry settings, shared DLLs, unrelated to the test procedures, have not been inadvertently overwritten by other software products.

Define Agile Users and Permissions

Use Agile Java Client to provide the application adapter testers with privileges they need to create and release test data.

Set Up the Agile Environment

Administrator settings affect only the way you enter data in Agile. It is a good idea to set the field lengths to match those of the target system to reduce possible data rejection; ChangeCAST will not automatically truncate any attribute as part of generating the transfer file. For more details, see Transferring Files on page 21.

Make sure the application adapter users finalize and review the profile before beginning testing. Name the profile library 'ChangeCAST.agx' and ensure it is in a specific location (by default, in the directory Agile\Agile 9.2.2.7\ChangeCAST).

Start ChangeCAST, select the profile, and verify all ChangeCAST settings, as described in the *Agile ChangeCAST User Guide (* "Working with Profiles"*)* and in Creating a Profile on page 23, before processing change orders through the application adapter. Then, enable the profile to audit and activate the profile for use.

Setting Up the Target System Test Environment

Verify Target System Accounts and Permissions

Set up user names, accounts, or privileges as needed on the Target host system to install and run the application adapter. Ensure all accounts refer to the test environment, and not the production environment.

Install the Application Adapter

Install the application adapter and review any installation errors to make sure the installation is correct, including any post-processor or data processor.

Verify FTP and Other Transfer Protocols

Before testing, make sure the FTP services are operable and all email or other communication servers are running.

Extract Sample Data from the Target System and Load into Agile

With the help of an Agile Solutions Consultant, use the sample data that you extract from the target system into an Agile test database. Use this database for development and integration testing. Changes made to these items and BOMs will identify data issues and flaws in the application adapter specification.

Creating Target System Environment Test Data

The test set provides a starting point to aid in the creation of a meaningful and useful test bed. Use the following tables to manually create and release change orders in Java Client:

- Item creation
- Item update
- BOM creation
- BOM update
- Combo change order

Each change order will generate one transfer file. Then process the files with the application adapter. Verify the results using the target system tables, reports, and utilities. In Agile, items can be parts or documents. Be sure to test multiple types of each. Create and release each of the following tests as individual change orders.

ECO/ MCO Number	Test Name	Procedure
	Item creation, single part	Create and release one new Item. Add the part with a description and a revision.
	Item creation, multiple parts	Create and release multiple new parts.
	Item creation, single document	Create and release one new document item.
	Item creation, multiple documents	Create and release multiple new documents.
	Item creation, default values	If the profile or application adapter supports default values, make sure these are populating the database.
	Item creation, conditional values	If the profile or application adapter supports conditional values, make sure these are populating the database.

Item Update Tests

The table below lists Item update tests.

ECO/ MCO number	Test name	Procedure
	Item update, description	Create a change order that updates the description of one item. Release the new revision.
	Item update, default values	Make sure that fields that the user should manage in the target system are not being overwritten with Agile data. Carefully review this in the application adapter data mapping tables.
	Item update, revision	Release new revision of one item.
	Item update, new and changed parts	On a single change order, change existing and create new parts.
	Item update, new and changed documents	On a single change order, change and create documents.
	Item update, default values	Update a default field for an item previously loaded in this test suite on the target system host. Create a change order that will modify that same item. Verify that defaults are not overwriting the changes made to the target databases.

The table below lists BOM Creation Tests.

ECO/ MCO number	Test name	Procedure
	BOM create, one component	Create/release new BOM with one new part.
	BOM create, multiple components, single level	Create/release new BOM with multiple parts but no subassemblies.
	BOM create, complex	Create several assemblies and interrelated subassemblies.
	BOM create, complex, with redlining	Create and release a BOM with multiple redlines, and redline an existing component or document to update the item attributes.
	BOM create, default values	Verify that default values defined in the template are populating the database.

BOM update tests

ECO/ MCO number	Test name	Procedure
	BOM update, new quantity per assembly	On one BOM, change the quantity on one component.
	BOM update, new item number	On one BOM, change the item number on one component.
	BOM update, use/swap item numbers	On one BOM, swap item numbers on multiple components.
	BOM update, additional reference designators	On one BOM, add a component with reference designators.
	BOM update, change reference designators	On one BOM, change reference designators.
	BOM update, renumber items	On an existing BOM, delete a BOM row in the middle of the assembly, then renumber the remaining items.
	BOM update, different number of reference designators	On a BOM that has been loaded into Agile from the target system, change a reference designator so it is much shorter or longer than in the previous revision.
	BOM update, data from reference designator is in notes field	On a BOM that has been loaded into Agile from the target system, change a reference designator so it has associated Agile Note data.
	BOM update, replacement component	On one BOM, replace a component with another component.
	BOM update, item deletion	On one BOM, delete/disable a component.

ECO/ MCO number	Test name	Procedure
	BOM update, complex	On one BOM, update/add/delete multiple components.
	BOM deletion	Delete all components for a BOM.
	BOM update, default values	Accessing a BOM, modify a field that was loaded with a default value when added to the databases. Change the BOM in Agile and release the change order. Verify that the default value does not overwrite the field that was just changed in the target system database.

Note In the BOM update tests, be sure to use BOMs loaded from the target system and BOMs created in Agile. Pay close attention to reference designator and effective date management between Agile and the target system.

The table below lists Combo Change Order Tests.

ECO/ MCO number	Test name	Procedure
	Combo change order, item/BOM create	Create/release multiple items; create/release BOM with multiple components redlined to add a component; change quantity, find number, and new/updated reference designators.
	Combo change order, update QPA, Component, and Item Number	On one BOM, change the quantity for one component, the component for an item, and the item number for a component.
	Combo change order, update multiple complex BOMs	On several BOMs, change several fields.
	Combo change order, add/update sequence logic	Create and release a change order where a BOM is added, another BOM is changed, then a third BOM is added. Name the assemblies so the change comes between the two additions: BOM1 — Added BOM2 — Changed BOM3 — Added
	Recursive BOM	Build a top-level assembly with components and at least one subassembly. In the subassembly call the top-level assembly.

Functional Tests

Include the following items in functional testing:

- Run critical business reports, utilities, and transactions. (Create work order, sales order, ship the product, run MRP against the product, make sure the product can be returned, run Cost Accounting reports against parts and BOMS loaded.)
- Run month-end jobs
- Run quarter-end jobs
- Run year-end jobs

Note It is imperative that you exercise all business functions of the target system fully to validate that the adapter is creating and updating all fields in the target system correctly.

Troubleshooting

Due to the nature of integrating systems, it is necessary to look at all the elements when troubleshooting a problem. Start with the data representation in Agile and the target system.

Data Representation

Data representation in the target system

Gather the current component information from the target system. This should be equivalent to the information used to change the change order in Agile (the prior revision). If they are not equivalent, determine which system contains the appropriate information. You need to make changes to the system with the errors. If the target system is in error, find out why it varies from the same revision in Agile; perhaps some commands need to be disallowed or some restrictions need to be deployed on the target system.

Data loading and current representation in Agile

If it turns out that Agile is in error, review the mapping used during the data load process. Make sure that the data has populated the correct fields in Agile. A new data load may be required if there are serious problems. You also need to make sure that this mapping was considered when the profile was developed.

Identification of Root Cause

Once you have a firm understanding of the data, then it is easy to determine the root cause of the problem. There are several places where a breakdown might occur.

What the intended change is and how the change order is entered

The following steps should help in determining why data is not being transferred properly.

Look at the change order in Agile Java Client:

- Does it have a new revision number on all affected items?
- Does it have an effective date on all affected items with new revisions?
- If the affected item is being changed, does it have both old and new revisions and effective and obsolete dates?
- What types of changes are being done? Changing item fields? Redlining BOMs? How are the changes being done?
- Are there other change orders that affect some or all of the information in the failing change order? In what order should you process the change orders? What is the release date and time for each of the affected change orders?

What are the profile mappings and file and transfer settings

Get a hard copy of the transfer file. Compare the output with the ChangeCAST settings.

- Does the transfer file have the correct file format (file structure, delimiters, data format, BOM row settings)?
- Does the transfer file correctly reflect the field mappings? Are all attributes accounted for?
- Are the transfer settings sending the files to the correct local or FTP location?

If any of these are in error, adjust the profile, reset the Set Transfer Flag and re-transmit the files (see “Formatting and Transferring Agile Data Files,” in the *Agile ChangeCAST User Guide*).

What is the data processor's functionality

If there is a data processor, get a hard copy of the output that interfaces with the application adapter if possible. Compare this to the input requirements of the application adapter. Correct any deviations.

What is the application adapter doing

Refer to the application adapter Design Specifications to see what is expected of the application adapter. Use the target system’s reports, online screens, or database listings to identify what the application adapter is actually doing. Make corrections where required.

Chapter 7

Data Processors

This chapter includes the following:

A gap may exist between the format of a transfer file and the file format required by a currently existing import-validation agent (IVA) or application adapter. This chapter provides information about data processors and how they are used during system integration.

Why a Data Processor?

A data processor residing on the host system can provide the additional processing needed to provide the expected formats. For example, the target system may require date conversion to provide an integer format.

In some cases you need to separate records based on whether they would initiate add, change or delete logic. ChangeCAST can generate files for each of the tables, but further separation requires a data processor.

For parts, you need to map an additional field containing the old revision for the part number. The presence of an old revision and a new revision indicate that the part is to be changed.

Deleting parts from Agile is not recommended. However, if you need to support this, the data processor could use the lifecycle phase (Inactive, Obsolete, or perhaps a custom Delete phase in Agile Web Client) to determine whether you need to delete the part.

Another case where a data processor can help is to derive net changes. BOM row output for changed rows will generate two records in the ChangeCAST transfer file.

Parent number	Child number	Find number	Quantity per assembly	Effective date	Obsolete date	Reference designator
GF-2000-00	GF-2000-01	10	6	##	3/12/1998	SP10-15
GF-2000-00	GF-2000-01	10	4	3/12/1998	##	SP10-13

The target import tool may need only the net change of the data shown above in a single row.

For example:

Net change in a single record

Parent number	Child number	Find number	Quantity per assembly	Effective date	Obsolete date	Reference designator
GF-2000-00	GF-2000-01	10	4	3/12/1998	##	SP10-13

Note that the information must be derived from the second of the two records shown in the first example. This would be done in a data processor.

Incorporating a Data Processor into the Integration Process

On systems using a data processor, you can locate the data processor on the target system, the Agile server, or another system on the network in line with the Agile server and the target system.

ChangeCAST also provides a way to launch a local application adapter after transferring changes. For details, see Application Adapter Tab on page 41.

Appendix A

ChangeCAST Macro Language

This Appendix includes the following:

Use Macros to provide conditional processing of field data.

The macro language syntax consists of functions, system variables, and standard C style expression syntax. You enter a macro in the Map Condition dialog box, described in Chapter 3 of the *Agile ChangeCAST User Guide.* Click the Test button to verify its syntax. A macro starts with an equal sign and may contain logical expressions, mathematical expressions, or string constants.

Example:

```
=1 returns the number 1.
="Text string" returns the string “Text string”.
=1==2 returns FALSE.
```

Mathematical Operators

Mathematical operators can operate only on numeric values.

*	Multiplication
/	Division
+	Addition
-	Subtraction

Example:

=1+2*3	returns the number 7.

Logical Operators

Most logical operators can operate only on numeric values (not strings or dates). Equal to and Not equal to can also operate on strings; both values must be strings.

==	Equal to

!=	Not equal to
!	Not
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Example:

=1==2 returns 0, which is equal to the logical FALSE.

Other Symbols

(	Open parenthesis
)	Close parenthesis

Functions

Functions return a value. A value may be a text string or a number. Logical functions return a number, where 0 is equal to logical FALSE, and 1 is equal to logical TRUE.

The ChangeCAST macro language does not fully support decimal numbers in scientific notation (for example, = 1e-1, = 5.15e-6) or numbers greater than 1e22. Also, several macro functions, such as LEN and other string functions, do not work with double-byte character sets. For more information, contact Agile Software Technical Support.

AND(logical1,logical2,…)

Parameters

Logical1	First logical expression to test.
Logical2	Second logical expression to test.

This function returns True if all its arguments are TRUE; FALSE if one or more arguments is FALSE.

Examples:

=AND(ISTEXT(1), ISTEXT("A")) returns a FALSE (0) because 1 is not equal to “1”.

=AND(ISTEXT(1), ISTEXT(2))	returns a FALSE (0) because both logicals return a FALSE response.
=AND(ISTEXT("A"), ISTEXT("B"))	returns a TRUE (1) because both logicals return a TRUE response.

CONCATENATE(text1,text2,…)

Parameters

text1 First text string.

text2 Second text string to concatenate with the first.

This function concatenates the text parameters and returns a string.

Example:

=CONCATENATE("first ", "last")	returns “first last”.

Note The \ character can be used as an escape character for the double quotation marks: **“Hello \”there\””** produces **Hello “there”**. You can also end the string with a backslash by using the \ to escape itself. For example, **“Backslash \\”** produces **Backslash **.

DATEVALUE(date_text)

Parameters

date_text The date string

Takes a string and converts it to an internal date value (time_t type). That date value can then be used by FORMATDATE and FORMATDATETIME. (Agile strongly recommends that you use this function with FORMATDATE and FORMATDATETIME.)

Note Date fields that contain a null or other non-date data are automatically filled with the current date. If you need a null value in such a field, use a macro like the following:

```
=IF(ISNULL([ChangeOrders.AffectedItemsTable.EffectiveDate:1079]),SYS_NU
LL,FORMATDATE(DATEVALUE([ChangeOrders.AffectedItemsTable.
EffectiveDate:1079]),"yyyy-dd-MMMM"))
```

FIXED(number,decimals)

Parameters

number The number to round and convert to text. This number can have up to 16 digits. If the number has more than 16 digits, it will be shown with some zeros in the decimal part as result of this function. For example: FIXED(21134567.222222222222222, 15) returns the value 21134567.222222222000000.

decimals The number of digits to the right of the decimal point. Only positive numbers are allowed for the decimals parameter.

Rounds a number to the specified number of decimals, formats the number in decimal format using a period and commas, and returns the result as text.

Example:

=FIXED(1234.567, 1)	returns “1234.6”

FORMATDATE(date_value, format)

Parameters

date_value The date/time value.

format The Windows style date format.

This function formats the date value according to the date format given in the parameter and returns a string. The date format is of the form d, dd, M, MM, MMM, MMMM, yyyy.

Examples

=FORMATDATE(TODAY(), "dd-MMM-yyyy")	returns a date value in a form similar to this: “28-Jan-1998”
=FORMATDATE(DATEVALUE([Parts.General Items Attribute:Rev Release Date:1016]), "yyyy-MMM-dd")	returns “1998-Jan-28”.

FORMATDATETIME(datetime_value, dateformat, timeformat)

Parameters

date_value The date/time value.

dateformat The Windows style date format.

timeformat The Windows style time format.

This function formats the date and time value according to the date and time formats given in the parameter and returns a string. The time is formatted after the date. The date format is the same as that for the FORMATDATE function. The time format can be h, hh, H, HH, m, mm, s, ss, t and tt.

Example:

```
=FORMATDATETIME(DATEVALUE([Change Orders.Affected Items Table.Obsolete
Date:1078]), "yyyy-MMM-dd", "HH:mm:ss")
```

Returns the value 1998-JAN-01 15:00:01 (24-hour clock). Ensure you indicate both time and date formats in the **Data Format** tab of the File Layout Settings dialog box.

FORMATNUMBER(number,format)

Parameters

number The number to format.

format The C-style format.

This function returns a text string for the number, formatted according to the C style formatting syntax and supports the following formats.: %d, %f, %e, %E, %g, %G.

Examples

=FORMATNUMBER(12.34, "%09.3f")	returns a number in this form: “00012.340”.
=FORMATNUMBER(123, "%5d")	returns “00123”.
=FORMATNUMBER(123, "%<space>5d")	returns “<space><space>123”.

Note Be sure to specify the appropriate data type. For example, if you divide two integers and specify an integer format, any fractional components of the result will not appear. Type float is required for fractional results.

IF(logical_test_value,value_if_true,value_if_false)

Parameters

logical_test_valueThe logical expression.

value_if_true Value to return if the logical expression evaluates to TRUE.

value_if_false Value to return if the logical expression evaluates to FALSE.

This function will evaluate the logical expression, and if the expression is TRUE, it returns the value in the value_if_true parameter. If the logical expression is false, it returns the value in the value_if_false parameter.

Example:

=IF([Change Orders.Affected Items Table.Lifecycle Phase:1057]==

"Inactive", "P", "I")

returns “I”.

ISBLANK(field)

Parameters

field The field being tested.

This function returns TRUE (1) if the field is blank. Otherwise, it returns FALSE (0).

Example:

=IF(ISBLANK([Parts.General Items Attributes.Rev:1014]), "Invalid", "Valid")

returns “Invalid” if a blank revision has been supplied, and “Valid” if a non-blank revision is supplied.

ISNULL(field)

Parameters

field The field being tested.

This function returns TRUE if the field has the SQL NULL value. Otherwise, it returns FALSE.

Example:

```
=IF(ISNULL([Parts.General Item Attributes.Number:1001]), "Invalid",
"Valid")
```

returns “Invalid” if the field has the SQL NULL value, “Valid” if it does not.

ISNUMBER(value)

Parameters

value The string, number, or field to test.

This function returns TRUE if the field value is a number. Otherwise, it returns FALSE. Valid numeric characters are:

- 0-9
- .
- +
- -

You can use the characters E and e with ISNUMBER.

Examples:

=ISNUMBER(1)	returns TRUE (1).
=ISNUMBER("1")	returns FALSE (0).
=IF(ISNUMBER([Part.General Item Attributes.Rev:1014]), "Valid", "Invalid")	returns “Invalid” if the revision has any non-numeric characters in it, and “Valid” if the revision has only numeric characters.
=ISNUMBER(1.3e2)	returns TRUE (or 1).

ISTEXT(value)

Parameters

value Text, number or field to test.

This function returns TRUE if the field contains one or more alpha-numeric characters. Otherwise, it returns FALSE.

Examples:

```
=ISTEXT(1)
```

returns FALSE (0).

```
=ISTEXT("1")
```

returns TRUE (1).

```
=IF(ISTEXT([Part.General Item Attributes.Rev:1014]), "Valid",
"Invalid")
```

returns “Valid” if the revision has any alpha characters in it, and “Invalid” if the revision has only numeric values in it.

LEFT(text, num_chars)

Parameters

text Text string to extract from.

num_chars Number of characters from the left to return.

This function returns the first num_chars characters of the text string. The text parameter supports integer constants up to 1e10. The num_chars parameter supports values of up to 1e18.

Examples

=LEFT("First five paragraphs", 5)	returns the string “First”.
=IF(LEFT([Change Orders.Affected Items Table.Lifecycle Phase:1057], 1)=="P", "Active", "Obsolete")	returns “Active” if the lifecycle phase is “Production”, “Pilot”, or “Prototype”. Otherwise, it returns “Obsolete”.

LEN(text)

Parameters

text The text string.

This function returns the length of the text string parameter.

Examples

=LEN("The text string")	returns the number 15.
=IF(LEN([Parts.General Item Attributes.Number:1001])==18, [Parts.General Item Attributes.Number:1001], CONCATENATE([Parts.General Item Attributes.Number:1001], "-00"))	returns the part number if the part number is 18 characters; otherwise, it returns the Part Number plus “-00” (123479807-00).

LOWER(text)

Parameters

text The text string to convert.

This function converts the text parameter to lowercase and returns the converted string.

Example:

=LOWER("LOWERCASE")	returns the string “lowercase”.

MID(text, start_num, num_chars)

Parameters

text The text string to examine.

start_num The first character. The first character in the text is in position 0.

num_chars The number of characters to examine after the start character.

This function locates and returns a part of the text string. The partial string starts *start_num* characters from the first character on the left, with the count starting at 0. The returned string includes the *num_chars* number of characters in the returned string. The text parameter supports integer constants up to 1e10.

Example:

=MID("THE MIDDLE OF THE SENTENCE.", 11, 2)	returns the string “of”.

NOT(logical)

Parameters

logical The logical expression to invert.

This function inverts the logical expression. If the logical expression is TRUE, the function returns FALSE and vice versa.

Example:

=NOT("TEST"=="TEST") returns FALSE (0) because they are exactly equal.

OR(logical1,logical2,...)

Parameters

logical1 First logical expression.

logical2 Second logical expression.

This function evaluates the logical expressions using the OR logical operator and returns TRUE or FALSE.

Example

```
=IF(OR([Change Orders.Affected Items Table.Lifecycle Phase:1057]==
"PRODUCTION",[Change Orders.Affected Items Table.Lifecycle
Phase:1057]=="production",[Change Orders.Affected Items Table.Lifecycle
Phase:1057]=="Production"), "Production", "Obsolete")
```

returns “Production” if any of the allowable versions of the word Production is entered. Otherwise, “Obsolete” will be returned.

REPLACE(old_text,start_num,num_chars,new_text)

Parameters

old_textThe original text string.

start_num Position from the left of the string to start the replacement. The offset position starts at 0.

num_chars The number of characters to replace.

new_text The string to replace the characters with.

This function replaces the text string passed in the old_text parameter with text passed in the new_text parameter. The characters in the old_text string to be replaced are specified by the start_num and num_chars parameter. The old_text parameter supports integer constants up to 1e10.

Example:

=REPLACE("This old man", 5, 3, "young") returns the string “This young man”.

RIGHT(text, num_chars)

Parameters

text The text string.

num_chars The number of characters from the right of the text string to return.

This function locates and returns a part of the text string. The partial string starts *start_num* characters from the first character on the right, and includes the characters from that point back to the right end of the string. The text parameter supports integer constants up to 1e10. The num_chars parameter supports values of up to 1e18.

Examples:

=RIGHT("A short string", 10)	returns the string “ort string”.
=RIGHT([Parts.General Item Attributes.Rev Release Date:1016], 4)	returns the four-character year of the Release Date (“1998”).

TODAY()

This function returns the current date. If TODAY() is used with FORMATDATE, the current date is returned; if TODAY() is used with FORMATDATETIME, the current date and time are returned.

Example:

=FORMATDATE(TODAY(), "dd-MMM-yyyy")	returns a date value in a form similar to this: “28-Jan-1998”

TRIM(text)

Parameters

text Text string to be trimmed.

This function removes leading and trailing white space from the text string. The text parameter supports integer constants up to 1e10.

Examples

```
 =TRIM("     leading white space     ")
```

returns the string "leading white space".

```
 =TRIM([Parts.General Item Attributes.Description:1002])
```

returns a part description with no leading or trailing white space.

UPPER(text)

Parameters

text The string to convert to uppercase.

This function converts a text string to uppercase.

Example:

=UPPER("uppercase")	returns the string “UPPERCASE”.

VALUE(text)

Parameters

text The string to convert to a number.

This function converts a text string that has a numeric value to an actual number.

Note If the number is less than 1e-6, this function returns a decimal only up to 6 decimal digits after the decimal point. For example, VALUE(“5.15e-6”) returns the value 0.000005. However, the real value is used internally.

Examples:

=VALUE("1234")	returns the number value 1234.
=VALUE("abc")	returns 0 (false).

System Variables

The following are system variables:

- **SYS_ACD** – This variable is similar to SYS_AMLACD, except that it applies to the BOM table and should be mapped only to the BOM table. It returns A if there are BOM redline additions, C if there are BOM redline changes, D if there are BOM redline deletions, and U if the BOM table is not redlined (that is, it is unmodified). See page 5-10 for a more detailed description.
- **SYS_AMLACD** – This variable is similar to SYS_ACD, except that it applies to the Manufacturer table and should be mapped only to the Manufacturer table. It returns A if there are Manufacturer redline additions, C if there are Manufacturer redline changes, D if there are Manufacturer redline deletions, and U if the Manufacturer table is not redlined (that is, it is unmodified).
- **SYS_HAS_PENDING_CHANGES** – This variable is TRUE (1) if the current BOM row has pending changes. A BOM row is determined to have pending changes if the affected item has pending changes.

- **SYS_HAS_PREVIOUS_REVISION** – This variable is TRUE (1) if the current BOM row has previous revisions. A BOM record has previous revisions if the Old Revisions field of the Affected Items page is not NULL or is not blank.
- **SYS_HAS_REVISIONS** – This variable is TRUE (1) if the current BOM row has revisions.
- **SYS_IS_ASSEMBLY** – This variable is TRUE (1) if an item is an assembly. A map to Parts.General Items Attributes.Number:1001 must be defined before you map a field to this special condition.
- **SYS_IS_CHANGED_RECORD** – This variable is TRUE (1) if the current BOM row is a change record.
- **SYS_IS_DELETED_RECORD** – This variable is TRUE (1) if the current BOM row is a deleted record.
- **SYS_IS_NEW_RECORD** – This variable is TRUE (1) if the current BOM row is a new record.
- **SYS_IS_UNMODIFIED_RECORD** – This variable is TRUE (1) if the current BOM row is an unmodified record.
- **SYS_LINECOUNT** – This variable returns the line number of the current mapped line.
- **SYS_NULL** – This variable returns the string representation for the SQL NULL value as specified by the user in the **Data Format** tab of the File Layout Settings dialog box.

  For example:

  - **SYS_NULL** returns "##" if the user has specified SQL NULL to be mapped to "##".
- **SYS_ROOTCHANGEID** – This variable returns a string. It returns CO if the root class is a Change Order or MO if the root class is a Manufacturer Order. It returns the string regardless of which table it is mapped to, and also regardless of the position in the table.

Note This variable replaces SYS_CLASSNAME. However, SYS_CLASSNAME will continue to be recognized to provide backward compatibility with existing mappings.

- **SYS_ROOTITEMID** – This variable returns the following:

PR	Part
DC	Document
--	When irrelevant (for a table where no item is mapped).

Scenarios

The following scenarios are meant to foster ideas and show practical examples of macro commands.

Scenario 1

Statement: Most assemblies in our database have a numeric revision. However, all components have alpha revisions but become assemblies at a later date and should have a number in the revision as well. This visibility is required only on the target system due to some reports that are generated.

Assessment: If the item refers to an assembly and the revision is an alpha character, prefix the revision with a “1”; otherwise, just use the passed revision.

Data: Assembly 0012245-01 has a revision of A. Assembly 0012205-00 has a revision of 1.

Expected Results: Assembly 0012245-01 should have a revision of 1A, and assembly 0012205-00 should have a revision of 1.

To accomplish this task, the following macro would be used in the Map Condition dialog box:

```
Map TABLE ONE.FIELD FOUR to
=IF(AND(SYS_IS_ASSEMBLY,ISTEXT([ChangeOrders.Affected Items Table.New
Rev:1056]),([Parts.BOM Table.BOM ID:-2]==[Parts.BOM Table.BOM Prior
ID:-1])),CONCATENATE(“1”,[ChangeOrders.Affected Items Table.New
Rev:1056]),[ChangeOrders.Affected Items Table.New Rev:1056])
```

The use of the AND in this situation allows for all the conditions to be tested (Is the part an assembly and is the re*vis*ion an alpha character, and is this the first occurrence of the item as an assembly?) and logic deployed based on the result. The SYS_IS_ASSEMBLY would return a TRUE (1) if the part were an assembly, and the ISTEXT would return a TRUE (1) if the revision were an Alpha (A-Z, a-z) character. The CONCATENATE command would be used only if the item in question were an assembly and the revision were alpha; it would return a 1X (where x is the passed revision). Otherwise, the revision (X) would simply be passed to the field. The IF evaluates the AND phrase and determines whether to deploy the CONCATENATE logic or just pass the revision.

Scenario 2

Statement: I need quantities rounded to the closest 100th of the number.

Assessment: Convert the quantity from a string to a numeric value, and then assign a 2-character decimal significance to the resultant number.

Data: Quantities range from “100” to “1.25” to “.0095”.

Expected Results: The quantity 100 should be formatted as “100.00”. The quantity 1.25 would be “1.25”, and the quantity .0095 would be “0.01”.

To achieve the desired results, use the following macro in the Map Condition dialog box:

```
Map BOM.QPA to =FIXED(VALUE([Parts.BOM Table.Qty:1035]),2)
```

The VALUE([Parts.BOM Table.Qty:1035]) would convert the quantity per assembly from a string to a numeric value. The FIXED command coupled with the 2 would round the result of the VALUE logical to the specified number of significant decimals. The trailing zeros are set in this version of the macro commands.

Scenario 3

Statement: Agile has a lot of different lifecycle phases, and we want to use them in Agile but the target system only knows A, O, and P.

Assessment: All lifecycle phases that begin with a “P” (Prototype, Pilot, Production) can be passed as an “A” Part. Obsolete parts will be sent with “O”, and Inactive parts will be sent with “P”.

Data:

Prototype = A
Pilot = A
Production = A
Obsolete = O
Inactive = P

Expected Results: See Data.

To generate the desired results, use the following macro in the Map Condition dialog box:

Table A-1:

```
Map ECOITEMS.LIFECYCLE_PHASE to =IF(LEFT([Change Orders.Affected Items Table.Lifecycle
Phase:1057],1)==“P”,”A”,IF([Change Orders.Affected Items Table.Lifecycle
Phase:1057]==“Inactive”,”P”,”O”))
```

The LEFT command selects the leftmost character from the lifecycle phase, which is then compared command to the letter “P”, returning “A” if it is a “P”. If it is not a “P”, and if the lifecycle phase is “Inactive”, it will return “P”; otherwise, it will return “O”.

Scenario 4

Statement: We do not always know what the part type is, but this is a required field on our target system. Can you just put a dummy value in when you pass the data to the host the first time? We will change it on the host later when we figure it out.

Assessment: If part type is null or blank and it is the initial release of this part, put “Misc” in the Part Type field; otherwise, just pass the value in the Part Type field.

Data: Part number 12345-00 has no part type and no old revision. Part number 12346-00 has no part type (blank) but has an old revision.

Expected Results: Part number 12345-00 will be transferred with a part type of “Misc”.

Part number 12346-00 will be transferred with a part type of blank.

The following would be used to generate the desired output:

TableA:2:

```
Map PART.PART_TYPE to =IF(AND(OR(ISBLANK([changeOrders.Affected Items Table.Old
Rev:1055]),ISNULL([Change Orders.Affected Items Table.Old Rev:1055])),OR(ISBLANK([Parts
General Item Attributes.Part Type;1081]),ISNULL([Parts General Item Attributes.Part
Type;1081])),”Misc”,[Parts General Item Attributes.Part Type;1081])
```

IF the Part Type ISBLANK OR ISNULL, AND the Old Rev ISNULL OR ISBLANK (initial release), THEN fill the Parts.PART TYPE field with “Misc”, otherwise (ELSE), send the Part Type that is there.

Scenario 5

Statement: We want to make sure that we can easily pick out the ECOs sent by Agile. We use a specific numbering system, so using the autonumber in Agile is not possible, but we want to place a C in front of all ECOs that are generated in Agile. New Agile ECO numbers will be entered as numeric only.

Assessment: From the data load, the Agile Certified Engineer realizes that the imported data was alphanumeric, but since they indicated that the ECO created in Agile will be all numeric, he can evaluate for entries that are numeric and add the prefix “C” to each.

Data: Imported ECOs look like the following: A12398-B or 55985-D. The ECO created in Agile will be numeric only and look like the following: 664529 or 990978.

Expected Results: Transferred ECO will look like the following: C664529 or C990978.

The following would be used to generate the desired output:

Table A-3:

```
Map ECO.CHANGE_NUMBER to =IF(ISNUMBER[Change Orders.General Change
Attributes.Number:1047]),CONCATENATE(“C”,[Change Orders.General Change
Attributes.Number:1047]),[Change Orders.General Change Attributes.Number:1047])
```

IF the ECO number ISNUMBER THEN CONCATENATE the ECO number with a “C”; otherwise (ELSE), just send the ECO number (this will allow auto generated numbers to be transferred, too).

Scenario 6

Statement: The application adapter programmer indicates that he is having a problem parsing because the data, though delimited, is not always the same length, especially in text fields, where there may be such data entered as double quotes. He mentions that if he knew the exact length of the field while he was parsing, his life would be a lot easier.

Assessment: First, evaluate the length of the field’s content, and then enter the actual field length followed by a colon and the fields content.

Data: BOM Notes fields sometimes have no data, while other records use all 1024 characters.

Expected Results: BOM Notes with no data would look like “0:”, while others might look like “28:Reference designators added.”

The following would be used to generate the desired output:

```
Map BOM.REMARKS to =CONCATENATE(LEN([Parts.BOM
Table.Notes:1036]),”:”,[Parts.BOM Table.Notes:1036])
```

CONCATENATE the LENgth of the Notes field with a colon (:) and the BOM Notes.

Scenario 7

Statement: “Our part numbers have both alpha and numeric data. Can we make sure that all alpha characters are converted to uppercase?”

Assessment: Set up Agile Java Client format for ALL and case to Upper to force uppercase for new data entry. For existing ECOs that may have mixed case, use a macro to convert all alpha characters in Part numbers to uppercase.

Data: Part number A4456b-01 and b3908c-d.

Expected Results: Part numbers will be transferred as follows: A4456B-01 and B3908C-D.

The following would be used to generate the desired output:

```
Map PART.PART_NUM to =UPPER([Parts.General Item
Attributes.Number:1001])
```

UPPER will convert all alpha characters in the Part number to uppercase.

Scenario 8

Statement: The description space for a part in the host system is limited, we want to strip the words 'Part Number' when it appears first in the description.

Assessment: If the first 11 characters in a description are 'Part Number' in any combination of upper- and lowercase, replace these 11 characters with nothing.

Data: The first 11 characters of many of the descriptions for parts are “Part Number” or “PART NUMBER” or “part number” (or some similar combination). For example: “Part number 2234564 is a child of 55456-00”

Expected Results: Description is now “2234564 is a child of 55456-00”.

The following would be used to generate the desired output:

```
Map PART.PART_DESC to =IF(LOWER(LEFT([Parts.General Item
Attributes.Description:1002],11))==”part number”,
REPLACE([Parts.General Item
Attributes.Description:1002],1,11,””),[Parts.General Item
Attributes.Description:1002])
```

IF the LOWERcase equivalent of the LEFT most 11 characters of the part's description is equal to “part number”, THEN REPLACE those 11 characters with nothing, otherwise (ELSE), pass the description as is.

Scenario 9

Statement: As a result of importing data into Agile, the Part Category field now contains two leading blank characters. The host system will not accept these two leading characters. We can correct this on an “as used” basis, but, as our BOMs are so big, we want to make sure that no part is accidentally missed. These blanks must be stripped prior to sending the data to the host.

Assessment: Eliminate the leading and trailing blanks in the Part Category field.

Data: “ Electrical”, “ Software ” , “ Mechanical ”, and “Electrical”, “Software”, “Mechanical”

Expected Results: “Electrical”, “Software”, and “Mechanical” will be transferred.

The following would be used to generate the desired output:

```
Map PART.PART_CATEGORY to =TRIM([Parts.General Item Attributes.Item
Category:1082])
```

TRIM all extraneous blanks from the item category field.

Scenario 10

Statement: The Oracle application adapter requires that the ACD flag be set as “1” for an addition, “2” for a change and “3” for a deletion. Agile sends an A, C, D, or U.

Assessment: Convert the SYS_ACD output to comply with the application adapter requirements.

Data: New BOM rows receive an ACD_Type of “A”, changed rows receive an ACD_Type of “C”, and deleted rows receive an ACD_Type of “D”. As we are using a delta orientation, unmodified (“U”) data is not sent.

Expected Results: A = 1, C = 2, D = 3.

The following would be used to generate the desired output:

```
Map ag_inventory_comps_interface ACD_TYPE to
=IF(SYS_ACD==”A”,”1”,(IF(SYS_ACD==”C”,”2”,IF(SYS_ACD==”D”,”3”,"")))
```

IF the SYS_ACD system variable is equal to “A”, THEN fill the field with a “1”. Otherwise (ELSE), IF the SYS_ACD system variable is equal to “C”, THEN fill the field with a “2”; otherwise, fill the field with a “3”.

Scenario 11

Statement: The MANMAN application adapter requires that a footer be written to the file with the following information:

# ECOHEAD Layout

ECOHEAD,ECO,10

ECOHEAD,RECORDS,5

# ECOHEAD Data

<ECO Number>

<Record Count>

(Record count in this situation is equal to file line count.)

Assessment: Create a table that will generate just the ECO number and the line count for the file.

Data: The transfer file length will populate the field value for line count.

Expected Results: The total line count, including the LINECOUNT line.

The following would be used to generate the desired output:

```
Map ECOHEAD.RECORDS to =SYS_LINECOUNT
```

SYS_LINECOUNT will count all the records up to and including the current line.

Appendix B

Passing Static Information

This Appendix includes the following:

In some instances, you need to pass static information to the target system. For example, the application adapter may need to know the DNS name or IP address, or the username and password of the ChangeCAST server, so that you can collect processing status in one central location. None of this information relates to an ECO, and transfer files are specifically generated to deal with ECO data. Being able to generate a table that has no reference to a change order data point is supported in ChangeCAST.

Procedures

Use the following procedure to pass the static information by creating a separate table. This example uses an FTP table to collect the FTP information for sending a status log back to the ChangeCAST server.

1. Modify the Attributes list. Create an FTP table and include fields like Host Name, User Name, Password, Target Path and any other data that will enable the Adapter to transfer a log entry or status report to the ChangeCAST server.
2. Edit the field mapping for the FTP table. Click each of the fields defined for the FTP table and click Map Condition. In the conditional section type the constant string value encased in double quotes for each of the fields you have defined (=“128.0.0.1”). The first field must be mapped in the following manner:

```
=IF([Change Orders.General Change Attributes.Number:1047]=="##",
"<host_name_ip>", "<host_name_ip>")
```

This mapping allows the table to be built regardless of the presence or absence of any Agile attributes; since you will always have an ECO number, you can always generate this table with its static data.

For example, a vertical format FTP table could be as follows:

FTP Layout
FTP, HOST_NAME_IP,10
FTP, USER_NAME, 16
FTP, PASSWORD, 8
FTP,TARGET_PATH,32
#FTP Data
<host_name_ip>

<user_name>
<password>
<target_path>

Worksheets

Item Worksheet

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
ITEM_ NUMBER		NOT NULL	TEXT	30	Yes	Upper			
PART TYPE			LIST	~	Yes	~	User Defined		
PART CATEGORY			LIST	~	No	~	User Defined		
DESCRIP-TION			MULTI-TEXT	100	Yes	Upper			
SIZE			LIST	~	No	~			
PRODUCT LINE(S)			MULTI-LIST	~	No	Mixed	User Defined		
REV INCORP DATE			DATE	~	No	~			
LIFECYCLE PHASE			LIST	~	Yes	~	User Defined		
REV			TEXT	20	Yes	Upper			
REV REL. DATE			DATE	~	No	~			
DATE01			DATE	~	No	~			
DATE02			DATE	~	No	~			

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
DATE03			DATE	~	No	~			
DATE04			DATE	~	No	~			
DATE05			DATE	~	No	~			
TEXT01			TEXT	50	No	Mixed			
TEXT02			TEXT	50	No	Mixed			
TEXT03			TEXT	50	No	Mixed			
TEXT04			TEXT	50	No	Mixed			
TEXT05			TEXT	50	No	Mixed			
TEXT06			TEXT	50	No	Mixed			
TEXT07			TEXT	50	No	Mixed			
TEXT08			TEXT	50	No	Mixed			
TEXT09			TEXT	50	No	Mixed			
TEXT10			TEXT	50	No	Mixed			
MULTI-TEXT10			MULTI-TEXT	256	No	Mixed			
MULTI-TEXT20			MULTI-TEXT	512	No	Mixed			
MULTI-TEXT30			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT31			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT32			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT33			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT34			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT35			MULTI-TEXT	1023	No	Mixed			
LIST01			LIST	~	No	~			
LIST02			LIST	~	No	~			

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
LIST03			LIST	~	No	~			
LIST04			LIST	~	No	~			
LIST05			LIST	~	No	~			
LIST06			LIST	~	No	~			
LIST07			LIST	~	No	~			
LIST08			LIST	~	No	~			
LIST09			LIST	~	No	~			
LIST10			LIST	~	No	~			
TEXT11			TEXT	50	No	Mixed			
TEXT12			TEXT	50	No	Mixed			
TEXT13			TEXT	50	No	Mixed			
TEXT14			TEXT	50	No	Mixed			
TEXT15			TEXT	50	No	Mixed			
TEXT16			TEXT	50	No	Mixed			
TEXT17			TEXT	50	No	Mixed			
TEXT18			TEXT	50	No	Mixed			
TEXT19			TEXT	50	No	Mixed			
TEXT20			TEXT	50	No	Mixed			
TEXT21			TEXT	50	No	Mixed			
TEXT22			TEXT	50	No	Mixed			
TEXT23			TEXT	50	No	Mixed			
TEXT24			TEXT	50	No	Mixed			
TEXT25			TEXT	50	No	Mixed			
LIST11			LIST	~	No	~			
LIST12			LIST	~	No	~			
LIST13			LIST	~	No	~			
LIST14			LIST	~	No	~			
LIST15			LIST	~	No	~			

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
LIST16			LIST	~	No	~			
LIST17			LIST	~	No	~			
LIST18			LIST	~	No	~			
LIST19			LIST	~	No	~			
LIST20			LIST	~	No	~			
LIST21			LIST	~	No	~			
LIST22			LIST	~	No	~			
LIST23			LIST	~	No	~			
LIST24			LIST	~	No	~			
LIST25			LIST	~	No	~			

BOM Worksheet

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when ACD
ITEM_NUMBER			TEXT	30	Yes	Upper	ITEM		
FIND_NUMBER			TEXT	~	Yes	Upper			
QUANTITY			TEXT	20	No	Upper			
DESCRIPTION			TEXT	100	No	Upper			
REV			TEXT	20	No	Upper			
REF DESIGNATOR			TEXT	~	No	Upper			
NOTES			TEXT	512	No	Mixed			

Name	Own (Y/N)	Valida-tions	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when ACD
BOM DATE 01			DATE	~	No	~			
BOMDATE 02			DATE	~	No	~			
BOM DATE03			DATE	~	No	~			
BOMDATE0 4			DATE	~	No	~			
BOMDATE0 5			DATE	~	No	~			
BOM TEXT01			TEXT	50	No	Mixed			
BOM TEXT02			TEXT	50	No	Mixed			
BOM TEXT03			TEXT	50	No	Mixed			
BOM TEXT04			TEXT	50	No	Mixed			
BOM TEXT05			TEXT	50	No	Mixed			
DATE01			DATE	~	No	~			
DATE02			DATE	~	No	~			
DATE03			DATE	~	No	~			
DATE04			DATE	~	No	~			
DATE05			DATE	~	No	~			
LIST01			LIST	~	No	~			
LIST02			LIST	~	No	~			
LIST03			LIST	~	No	~			
LIST04			LIST	~	No	~			
LIST05			LIST	~	No	~			
LIST06			LIST	~	No	~			
LIST07			LIST	~	No	~			

Name	Own (Y/N)	Valida-tions	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when ACD
LIST08			LIST	~	No	~			
LIST09			LIST	~	No	~			
LIST10			LIST	~	No	~			
LIST11			LIST	~	No	~			
LIST12			LIST	~	No	~			
LIST13			LIST	~	No	~			
LIST14			LIST	~	No	~			
LIST15			LIST	~	No	~			
LIST16			LIST	~	No	~			
LIST17			LIST	~	No	~			
LIST18			LIST	~	No	~			
LIST19			LIST	~	No	~			
LIST20			LIST	~	No	~			
LIST21			LIST	~	No	~			
LIST22			LIST	~	No	~			
LIST23			LIST	~	No	~			
LIST24			LIST	~	No	~			
LIST25			LIST	~	No	~			
MULTI-TEXT10			MULTI-TEXT	256	No	~			
MULTI-TEXT20			MULTI-TEXT	512	No	~			
MULTI-TEXT30			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT31			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT32			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT33			MULTI-TEXT	1023	No	Mixed			
MULTI-TEXT34			MULTI-TEXT	1023	No	Mixed			

Name	Own (Y/N)	Valid a-tions	User Inter face	Le ngt h	Requ ired	Cases	Depend encies or affected tables	Transfer red to ERP field name	Update when ACD
MULTI-TEXT35			MULTI-TEXT	10 23	No	Mixed			
TEXT01			TEXT	50	No	Mixed			
TEXT02			TEXT	50	No	Mixed			
TEXT03			TEXT	50	No	Mixed			
TEXT04			TEXT	50	No	Mixed			
TEXT05			TEXT	50	No	Mixed			
TEXT06			TEXT	50	No	Mixed			
TEXT07			TEXT	50	No	Mixed			
TEXT08			TEXT	50	No	Mixed			
TEXT09			TEXT	50	No	Mixed			
TEXT10			TEXT	50	No	Mixed			
TEXT11			TEXT	50	No	Mixed			
TEXT12			TEXT	50	No	Mixed			
TEXT13			TEXT	50	No	Mixed			
TEXT14			TEXT	50	No	Mixed			
TEXT15			TEXT	50	No	Mixed			
TEXT16			TEXT	50	No	Mixed			
TEXT17			TEXT	50	No	Mixed			
TEXT18			TEXT	50	No	Mixed			
TEXT19			TEXT	50	No	Mixed			
TEXT20			TEXT	50	No	Mixed			
TEXT21			TEXT	50	No	Mixed			
TEXT22			TEXT	50	No	Mixed			
TEXT23			TEXT	50	No	Mixed			
TEXT24			TEXT	50	No	Mixed			
TEXT25			TEXT	50	No	Mixed			

Change Order Worksheet

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
CHANGE NUMBER			TEXT	30	Yes	Upper		
CHANGE TYPE			LIST	~	Yes	~	System Table	
CHANGE CATEGORY			LIST	~	Yes	~	User Defined	
STATUS			LIST	~	Yes	~	System Table	
REASON CODE			LIST	~	Yes	~	User Defined	
ORIGINATOR			LIST	~	Yes	~	System Tables	
CHANGE ANALYST			LIST	~	No	~	System Tables	
DATE ORIGINATED			DATE		Yes	~		
DATE RELEASED			DATE		Yes	~		
DESCRIPTION FOR CHANGE			MULTI-TEXT	1023	Yes	Mixed		
REASON FOR CHANGE			MULTI-TEXT	1023	Yes	Mixed		
PRODUCT LINE(S)			MULTI-LIST	~	No	~	User Defined	
TRANSFERRED			TEXT	50	No	Mixed		
WORK-FLOW			TEXT		Yes	Mixed	Default or User Defined	
DATE01			DATE		No	~		
DATE02			DATE		No	~		
DATE03			DATE		No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
DATE04			DATE		No	~		
DATE05			DATE		No	~		
TEXT01			TEXT	50	No	Mixed		
TEXT02			TEXT	50	No	Mixed		
TEXT03			TEXT	50	No	Mixed		
TEXT04			TEXT	50	No	Mixed		
TEXT05			TEXT	50	No	Mixed		
TEXT06			TEXT	50	No	Mixed		
TEXT07			TEXT	50	No	Mixed		
TEXT08			TEXT	50	No	Mixed		
TEXT09			TEXT	50	No	Mixed		
TEXT10			TEXT	50	No	Mixed		
TEXT11			TEXT	50	No	Mixed		
TEXT12			TEXT	50	No	Mixed		
TEXT13			TEXT	50	No	Mixed		
TEXT14			TEXT	50	No	Mixed		
TEXT15			TEXT	50	No	Mixed		
TEXT16			TEXT	50	No	Mixed		
TEXT17			TEXT	50	No	Mixed		
TEXT18			TEXT	50	No	Mixed		
TEXT19			TEXT	50	No	Mixed		
TEXT20			TEXT	50	No	Mixed		
TEXT21			TEXT	50	No	Mixed		
TEXT22			TEXT	50	No	Mixed		
TEXT23			TEXT	50	No	Mixed		
TEXT24			TEXT	50	No	Mixed		
TEXT25			TEXT	50	No	Mixed		
LIST01			LIST	~	No	~		
LIST02			LIST	~	No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
LIST03			LIST	~	No	~		
LIST04			LIST	~	No	~		
LIST05			LIST	~	No	~		
LIST06			LIST	~	No	~		
LIST07			LIST	~	No	~		
LIST08			LIST	~	No	~		
LIST09			LIST	~	No	~		
LIST10			LIST	~	No	~		
LIST11			LIST	~	No	~		
LIST12			LIST	~	No	~		
LIST13			LIST	~	No	~		
LIST14			LIST	~	No	~		
LIST15			LIST	~	No	~		
LIST16			LIST	~	No	~		
LIST17			LIST	~	No	~		
LIST18			LIST	~	No	~		
LIST19			LIST	~	No	~		
LIST20			LIST	~	No	~		
LIST21			LIST	~	No	~		
LIST22			LIST	~	No	~		
LIST23			LIST	~	No	~		
LIST24			LIST	~	No	~		
LIST25			LIST	~	No	~		
MULTI-TEXT10			MULTI-TEXT	256	No	~		
MULTI-TEXT20			MULTI-TEXT	512	No	~		
MULTI-TEXT30			MULTI-TEXT	1023	No	~		
MULTI-TEXT31			MULTI-TEXT	1023	No	Mixed		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
MULTI-TEXT32			MULTI-TEXT	1023	No	Mixed		
MULTI-TEXT33			MULTI-TEXT	1023	No	Mixed		
MULTI-TEXT34			MULTI-TEXT	1023	No	Mixed		
MULTI-TEXT35			MULTI-TEXT	1023	No	Mixed		
MULTI-LIST01			MULTI-LIST	~	No	~		
MULTI-LIST02			MULTI-LIST	~	No	~		
MULTI-LIST03			MULTI-LIST	~	No	~		

Affected Items Worksheet

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
STOCK			LIST	~	No	~			
NEW REV			TEXT	20	No	Upper			
OBSOLETE DATE			DATE	~	No	~			
EFFECTIVE DATE			DATE	~	Yes	~			
ON ORDER			LIST	~	No	~			
FIELD			LIST	~	No	~			
ITEM NUMBER			TEXT	30	Yes	Upper			

Name	Own (Y/N)	Vali dati ons	User Inter face	Len gth	Requ ired	Cases	Depend encies or affected tables	Transfe rred to ERP field name	Up date when (ACD)
OLD REV			TEXT	20	No	Upper			
FINISHED GOODS			LIST	~	No	~			
CHANGE FUNCTION			LIST	~	No	~	User Defined		
LIFECYCLE PHASE			LIST	~	Yes	~	System Tables		
WORK IN PROGRESS			LIST	~	No	~			
DESCRIP-TION			MULTI-TEXT	100	No	Upper			
DATE01			DATE	~	No	~			
DATE02			DATE	~	No	~			
DATE03			DATE	~	No	~			
DATE04			DATE	~	No	~			
DATE05			DATE	~	No	~			
DATE06			DATE	~	No	~			
DATE07			DATE	~	No	~			
DATE08			DATE	~	No	~			
DATE09			DATE	~	No	~			
DATE10			DATE	~	No	~			
DATE11			DATE	~	No	~			
DATE12			DATE	~	No	~			
DATE13			DATE	~	No	~			
DATE14			DATE	~	No	~			
DATE15			DATE	~	No	~			
DATE16			DATE	~	No	~			
DATE17			DATE	~	No	~			

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Update when (ACD)
DATE18			DATE	~	No	~			
DATE19			DATE	~	No	~			
DATE20			DATE	~	No	~			
LOC06			LIST	~	No	~			
LOC07			LIST	~	No	~			
LOC08			LIST	~	No	~			
LOC09			LIST	~	No	~			
LOC10			LIST	~	No	~			
LIST01			LIST	~	No	~			
LIST02			LIST	~	No	~			
LIST03			LIST	~	No	~			
LIST04			LIST	~	No	~			
LIST05			LIST	~	No	~			
LIST06			LIST	~	No	~			
LIST07			LIST	~	No	~			
LIST08			LIST	~	No	~			
LIST09			LIST	~	No	~			
LIST10			LIST	~	No	~			
LIST11			LIST	~	No	~			
LIST12			LIST	~	No	~			
LIST13			LIST	~	No	~			
LIST14			LIST	~	No	~			
LIST15			LIST	~	No	~			
LIST16			LIST	~	No	~			
LIST17			LIST	~	No	~			
LIST18			LIST	~	No	~			
LIST19			LIST	~	No	~			
LIST20			LIST	~	No	~			

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name	Up date when (ACD)
LIST21			LIST	~	No	~			
LIST22			LIST	~	No	~			
LIST23			LIST	~	No	~			
LIST24			LIST	~	No	~			
LIST25			LIST	~	No	~			
TEXT01			TEXT	50	No	Mixed			
TEXT02			TEXT	50	No	Mixed			
TEXT03			TEXT	50	No	Mixed			
TEXT04			TEXT	50	No	Mixed			
TEXT05			TEXT	50	No	Mixed			

Manufacturer Orders

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
CATEGORY			LIST	~	Yes	~		
SUBCLASS			LIST	~	Yes	~		
OWNER			LIST	~	No	~		
CREATE_ DATE			DATE	~	Yes	~		
RELEASE _ DATE			DATE	~	Yes	~		
DESCRIPTION			TEXT	1023	Yes	~		
CHANGE_ NUMBER			TEXT	30	Yes	~		
ORIGINA-			MULTI-	~	Yes	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
TOR			LIST					
PRODUCT _ LINES			MULTI-LIST	~	No	~		
REASON_ CODE			LIST	~	Yes	~		
REASON			MULTI-TEXT	1023	Yes	~		
STATUS			LIST	~	Yes	~		
TRANS-FERRED			TEXT	30	50	~		
WORKFLO W			TEXT		Yes	Mixed	Default or User Defined	
DATE01			DATE	~	No	~		
DATE02			DATE	~	No	~		
DATE03			DATE	~	No	~		
DATE04			DATE	~	No	~		
DATE05			DATE	~	No	~		
LIST01			LIST	~	No	~		
LIST02			LIST	~	No	~		
LIST03			LIST	~	No	~		
LIST04			LIST	~	No	~		
LIST05			LIST	~	No	~		
LIST06			LIST	~	No	~		
LIST07			LIST	~	No	~		
LIST08			LIST	~	No	~		
LIST09			LIST	~	No	~		
LIST10			LIST	~	No	~		
LIST11			LIST	~	No	~		
LIST12			LIST	~	No	~		
LIST13			LIST	~	No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
LIST14			LIST	~	No	~		
LIST15			LIST	~	No	~		
LIST16			LIST	~	No	~		
LIST17			LIST	~	No	~		
LIST18			LIST	~	No	~		
LIST19			LIST	~	No	~		
LIST20			LIST	~	No	~		
LIST21			LIST	~	No	~		
LIST22			LIST	~	No	~		
LIST23			LIST	~	No	~		
LIST24			LIST	~	No	~		
LIST25			LIST	~	No	~		
TEXT01			TEXT	50	No	Mixed		
TEXT02			TEXT	50	No	Mixed		
TEXT03			TEXT	50	No	Mixed		
TEXT04			TEXT	50	No	Mixed		
TEXT05			TEXT	50	No	Mixed		
TEXT06			TEXT	50	No	Mixed		
TEXT07			TEXT	50	No	Mixed		
TEXT08			TEXT	50	No	Mixed		
TEXT09			TEXT	50	No	Mixed		
TEXT10			TEXT	50	No	Mixed		
TEXT11			TEXT	50	No	Mixed		
TEXT12			TEXT	50	No	Mixed		
TEXT13			TEXT	50	No	Mixed		
TEXT14			TEXT	50	No	Mixed		
TEXT15			TEXT	50	No	Mixed		
TEXT16			TEXT	50	No	Mixed		
TEXT17			TEXT	50	No	Mixed		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
TEXT18			TEXT	50	No	Mixed		
TEXT19			TEXT	50	No	Mixed		
TEXT20			TEXT	50	No	Mixed		
TEXT21			TEXT	50	No	Mixed		
TEXT22			TEXT	50	No	Mixed		
TEXT23			TEXT	50	No	Mixed		
TEXT24			TEXT	50	No	Mixed		
TEXT25			TEXT	50	No	Mixed		

Manufacturer Orders Affected Items

Name	Own (Y/N)	Valida-tions	User Interface	Length	Required	Cases	Dependencies or affected tables	Trans-ferred to ERP field name
ITEM NUMBER			TEXT	30	Yes	~		
DESCRIPTION			MULTI-TEXT	100	Yes	~		
LIFECYCLE PHASE			LIST	~	Yes			
DATE01			DATE	~	No	~		
DATE02			DATE	~	No	~		
DATE03			DATE	~	No	~		
DATE04			DATE	~	No	~		
DATE05			DATE	~	No	~		
DATE06			DATE	~	No	~		
DATE07			DATE	~	No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
DATE08			DATE	~	No	~		
DATE09			DATE	~	No	~		
DATE10			DATE	~	No	~		
DATE11			DATE	~	No	~		
DATE12			DATE	~	No	~		
DATE13			DATE	~	No	~		
DATE14			DATE	~	No	~		
DATE15			DATE	~	No	~		
DATE16			DATE	~	No	~		
DATE17			DATE	~	No	~		
DATE18			DATE	~	No	~		
DATE19			DATE	~	No	~		
DATE20			DATE	~	No	~		
DISPOSITION 01			LIST	~	No	~		
DISPOSITION 02			LIST	~	No	~		
DISPOSITION 03			LIST	~	No	~		
DISPOSITION004			LIST	~	No	~		
DISPOSITION 05			LIST	~	No	~		
DISPOSITION			LIST	~	No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
06								
DISPOSITION 07			LIST	~	No	~		
DISPOSITION 08			LIST	~	No	~		
DISPOSITION 09			LIST	~	No	~		
DISPOSITION 10			LIST	~	No	~		
LIST01			LIST	~	No	~		
LIST02			LIST	~	No	~		
LIST03			LIST	~	No	~		
LIST04			LIST	~	No	~		
LIST05			LIST	~	No	~		
LIST06			LIST	~	No	~		
LIST07			LIST	~	No	~		
LIST08			LIST	~	No	~		
LIST09			LIST	~	No	~		
LIST10			LIST	~	No	~		
LIST11			LIST	~	No	~		
LIST12			LIST	~	No	~		
LIST13			LIST	~	No	~		
LIST14			LIST	~	No	~		
LIST15			LIST	~	No	~		

Name	Own (Y/N)	Valida-tions	User Interface	Length	Required	Cases	Dependencies or affected tables	Trans-ferred to ERP field name
LIST16			LIST	~	No	~		
LIST17			LIST	~	No	~		
LIST18			LIST	~	No	~		
LIST19			LIST	~	No	~		
LIST20			LIST	~	No	~		
LIST21			LIST	~	No	~		
LIST22			LIST	~	No	~		
LIST23			LIST	~	No	~		
LIST24			LIST	~	No	~		
LIST25			LIST	~	No	~		
TEXT01			TEXT	50	No	Mixed		
TEXT02			TEXT	50	No	Mixed		
TEXT03			TEXT	50	No	Mixed		
TEXT04			TEXT	50	No	Mixed		
TEXT05			TEXT	50	No	Mixed		

Manufacturer

Name	Own (Y/N)	Valida-tions	User interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
ADDRESS			MULTI-TEXT	1023	Yes	~		

Name	Own (Y/N)	Valida-tions	User interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
CITY			TEXT	128	Yes	~		
CONTACT			TEXT	128	Yes	~		
COUNTRY			LIST	~	Yes	~		
EMAIL			TEXT	128	Yes	~		
FAX			TEXT	128	Yes	~		
MFR TYPE			LIST	~	Yes	~		
NAME			TEXT	128	Yes	~		
PHONE			TEXT	128	Yes	~		
POSTAL CODE			TEXT	128	Yes	~		
STATE			LIST	~	Yes	~		
STATUS			LIST	~	Yes	~		
URL			TEXT	255	Yes	~		
DATE01			DATE	~	No	~		
DATE02			DATE	~	No	~		
DATE03			DATE	~	No	~		
DATE04			DATE	~	No	~		
DATE05			DATE	~	No	~		
LIST01			LIST	~	No	~		
LIST02			LIST	~	No	~		
LIST03			LIST	~	No	~		
LIST04			LIST	~	No	~		
LIST05			LIST	~	No	~		
LIST06			LIST	~	No	~		
LIST07			LIST	~	No	~		
LIST08			LIST	~	No	~		
LIST09			LIST	~	No	~		

Name	Own (Y/N)	Validations	User interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
LIST10			LIST	~	No	~		
LIST11			LIST	~	No	~		
LIST12			LIST	~	No	~		
LIST13			LIST	~	No	~		
LIST14			LIST	~	No	~		
LIST15			LIST	~	No	~		
LIST16			LIST	~	No	~		
LIST17			LIST	~	No	~		
LIST18			LIST	~	No	~		
LIST19			LIST	~	No	~		
LIST20			LIST	~	No	~		
LIST21			LIST	~	No	~		
LIST22			LIST	~	No	~		
LIST23			LIST	~	No	~		
LIST24			LIST	~	No	~		
LIST25			LIST	~	No	~		
MULTI-TEXT10			MULTI - TEXT	1023	No	~		
MULTI-TEXT20			MULTI - TEXT	1023	No	~		
MULTI-TEXT30			MULTI - TEXT	1023	No	~		
MULTI-TEXT31			MULTI - TEXT	1023	No	~		
MULTI-TEXT32			MULTI - TEXT	1023	No	~		
MULTI-TEXT33			MULTI - TEXT	1023	No	~		
MULTI-TEXT34			MULTI - TEXT	1023	No	~		
MULTI-TEXT35			MULTI - TEXT	1023	No	~		

Name	Own (Y/N)	Valida-tions	User interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
NOTES			MULTI - TEXT	1023	No	~		
TEXT01			TEXT	50	No	Mixed		
TEXT02			TEXT	50	No	Mixed		
TEXT03			TEXT	50	No	Mixed		
TEXT04			TEXT	50	No	Mixed		
TEXT05			TEXT	50	No	Mixed		
TEXT06			TEXT	50	No	Mixed		
TEXT07			TEXT	50	No	Mixed		
TEXT08			TEXT	50	No	Mixed		
TEXT09			TEXT	50	No	Mixed		
TEXT10			TEXT	50	No	Mixed		
TEXT11			TEXT	50	No	Mixed		
TEXT12			TEXT	50	No	Mixed		
TEXT13			TEXT	50	No	Mixed		
TEXT14			TEXT	50	No	Mixed		
TEXT15			TEXT	50	No	Mixed		
TEXT16			TEXT	50	No	Mixed		
TEXT17			TEXT	50	No	Mixed		
TEXT18			TEXT	50	No	Mixed		
TEXT19			TEXT	50	No	Mixed		
TEXT20			TEXT	50	No	Mixed		
TEXT21			TEXT	50	No	Mixed		
TEXT22			TEXT	50	No	Mixed		
TEXT23			TEXT	50	No	Mixed		
TEXT24			TEXT	50	No	Mixed		
TEXT25			TEXT	50	No	Mixed		

Manufacturer Parts

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
MFR NAME			TEXT	128	Yes	~		
MFR PART NUMBER			TEXT	128	Yes	~		
MFR PART DESCRIPTION			TEXT	100	No			
SUBCLASS (Mfr Part Type)			LIST	~	Yes	~		
STATUS			LIST	~	Yes	~		
DATE01			DATE	~	No	~		
DATE02			DATE	~	No	~		
DATE03			DATE	~	No	~		
DATE04			DATE	~	No	~		
DATE05			DATE	~	No	~		
LIST01			LIST	~	No	~		
LIST02			LIST	~	No	~		
LIST03			LIST	~	No	~		
LIST04			LIST	~	No	~		
LIST05			LIST	~	No	~		
LIST06			LIST	~	No	~		
LIST07			LIST	~	No	~		
LIST08			LIST	~	No	~		
LIST09			LIST	~	No	~		
LIST10			LIST	~	No	~		
LIST11			LIST	~	No	~		
LIST12			LIST	~	No	~		
LIST13			LIST	~	No	~		
LIST14			LIST	~	No	~		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
LIST15			LIST	~	No	~		
LIST16			LIST	~	No	~		
LIST17			LIST	~	No	~		
LIST18			LIST	~	No	~		
LIST19			LIST	~	No	~		
LIST20			LIST	~	No	~		
LIST21			LIST	~	No	~		
LIST22			LIST	~	No	~		
LIST23			LIST	~	No	~		
LIST24			LIST	~	No	~		
LIST25			LIST	~	No	~		
MULTI-TEXT10			MULTI-TEXT	1023	No	~		
MULTI-TEXT20			MULTI-TEXT	1023	No	~		
MULTI-TEXT30			MULTI-TEXT	1023	No	~		
MULTI-TEXT31			MULTI-TEXT	1023	No	~		
MULTI-TEXT32			MULTI-TEXT	1023	No	~		
MULTI-TEXT33			MULTI-TEXT	1023	No	~		
MULTI-TEXT34			MULTI-TEXT	1023	No	~		
MULTI-TEXT35			MULTI-TEXT	1023	No	~		
NOTES			MULTI-TEXT	1023	No	~		
TEXT01			TEXT	50	No	Mixed		
TEXT02			TEXT	50	No	Mixed		
TEXT03			TEXT	50	No	Mixed		

Name	Own (Y/N)	Validations	User Interface	Length	Required	Cases	Dependencies or affected tables	Transferred to ERP field name
TEXT04			TEXT	50	No	Mixed		
TEXT05			TEXT	50	No	Mixed		
TEXT06			TEXT	50	No	Mixed		
TEXT07			TEXT	50	No	Mixed		
TEXT08			TEXT	50	No	Mixed		
TEXT09			TEXT	50	No	Mixed		
TEXT10			TEXT	50	No	Mixed		
TEXT11			TEXT	50	No	Mixed		
TEXT12			TEXT	50	No	Mixed		
TEXT13			TEXT	50	No	Mixed		
TEXT14			TEXT	50	No	Mixed		
TEXT15			TEXT	50	No	Mixed		
TEXT16			TEXT	50	No	Mixed		
TEXT17			TEXT	50	No	Mixed		
TEXT18			TEXT	50	No	Mixed		
TEXT19			TEXT	50	No	Mixed		
TEXT20			TEXT	50	No	Mixed		
TEXT21			TEXT	50	No	Mixed		
TEXT22			TEXT	50	No	Mixed		
TEXT23			TEXT	50	No	Mixed		
TEXT24			TEXT	50	No	Mixed		
TEXT25			TEXT	50	No	Mixed		

ERP Target Worksheet

Identify all tables and fields affected by the interface.

Table Name

Name	Own (Y/N)	Validations	Format (field type: N, A, All)	Length	Required	Cases (upper, lower, mixed)	Dependencies	Transferred from Agile attribute name

Appendix C

Manufacturer Parts and Manufacturer Objects

This Appendix includes the following:

To make information about Manufacturer Parts available to the target system, ensure that the data is visible and the format is proper in the transfer files so that the adapter can process it.

Agile Configuration

There are two areas where Manufacturer data must be made visible within Agile for the attributes to appear within ChangeCAST mapping.

Typically, all the general information for a manufacturer is visible upon implementation of Agile with an AML license.

On opening **Parts Class** node, we notice there is a **Manufacturers** node there as well. This allows the association of the Agile part to all acceptable Manufacturer Parts. These attributes also need to be made visible as these are the attributes that can be mapped through ChangeCAST.

Select attributes that you need to transfer to your target system, making each one visible. If a part is visible under Parts class and Manufacturers class or Manufacturer Parts class, you can map it to ChangeCAST.

ChangeCAST

Once the part's manufacturer data is visible in Agile, you can map it to the appropriate fields in the target system. In the ChangeCAST Attributes table, you need to define the name and description of the target system table and fields before you use them as part of the mapping.

Attributes

To define attributes:

1. Run ChangeCAST application.
2. Select the profile you wish to add Parts Manufacturer data to.
3. Choose **Mapping | Edit Attributes**.
4. When you complete your definitions, click **OK** to save your new attributes in the profile.

Note Before you enter field definitions, send the Manufacturer part information in a separate table from other objects such as Part or BOM or Manufacturer itself. This is because the part's manufacturer data essentially is treated like BOM data by Agile. Mixing manufacturer data with item-specific data will produce unpredictable results in the transfer file. It is therefore suggested that you create tables for each of the new objects you want to transfer to the target system (MFR, AML, and MFRPRT if desired).

A File Header table is shown below. This would allow the adapter to identify whether an ECO or an MCO is in the transfer packet.

Example of Attributes Table

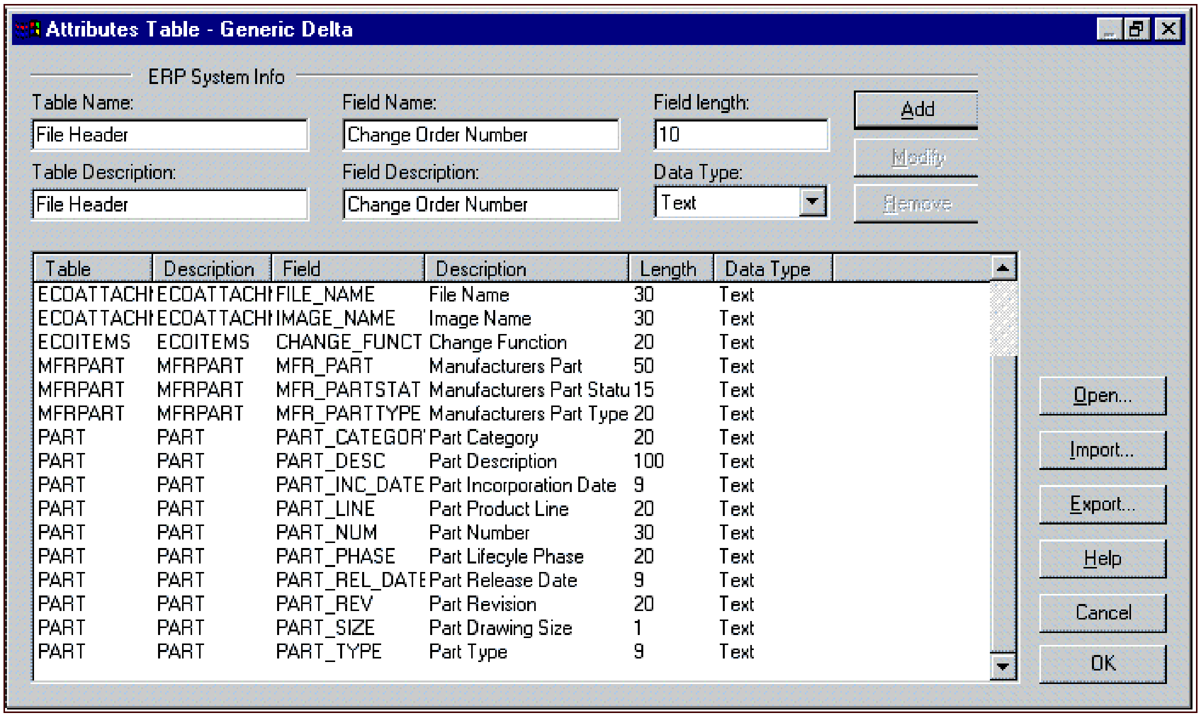


Table Name: File Header

Table Description: File Header

Field Name: Change Order Number

Field Description: Change Order Number

Field length: 10

Data Type: Text

Table	Description	Field	Description	Length	Data Type
ECOATTACHI	ECOATTACHI	FILE_NAME	File Name	30	Text
ECOATTACHI	ECOATTACHI	IMAGE_NAME	Image Name	30	Text
ECOITEMS	ECOITEMS	CHANGE_FUNCT	Change Function	20	Text
MFRPART	MFRPART	MFR_PART	Manufacturers Part	50	Text
MFRPART	MFRPART	MFR_PARTSTAT	Manufacturers Part Statu	15	Text
MFRPART	MFRPART	MFR_PARTTYPE	Manufacturers Part Type	20	Text
PART	PART	PART_CATEGOR	Part Category	20	Text
PART	PART	PART_DESC	Part Description	100	Text
PART	PART	PART_INC_DATE	Part Incorporation Date	9	Text
PART	PART	PART_LINE	Part Product Line	20	Text
PART	PART	PART_NUM	Part Number	30	Text
PART	PART	PART_PHASE	Part Lifecyle Phase	20	Text
PART	PART	PART_REL_DATE	Part Release Date	9	Text
PART	PART	PART_REV	Part Revision	20	Text
PART	PART	PART_SIZE	Part Drawing Size	1	Text
PART	PART	PART_TYPE	Part Type	9	Text

Note that the table AML in this example has been created with all the fields required for sending multiple acceptable manufacturers for a part to the target system. It also defines Manufacturer (MFR) and Manufacturer Parts (MFRPART) tables. Whenever a change order affects a part, the tables defined allow you to create and maintain data within Agile and sends it to the target system.

Mappings

Once you make the attribute definition, you can map the attributes to Agile attributes.

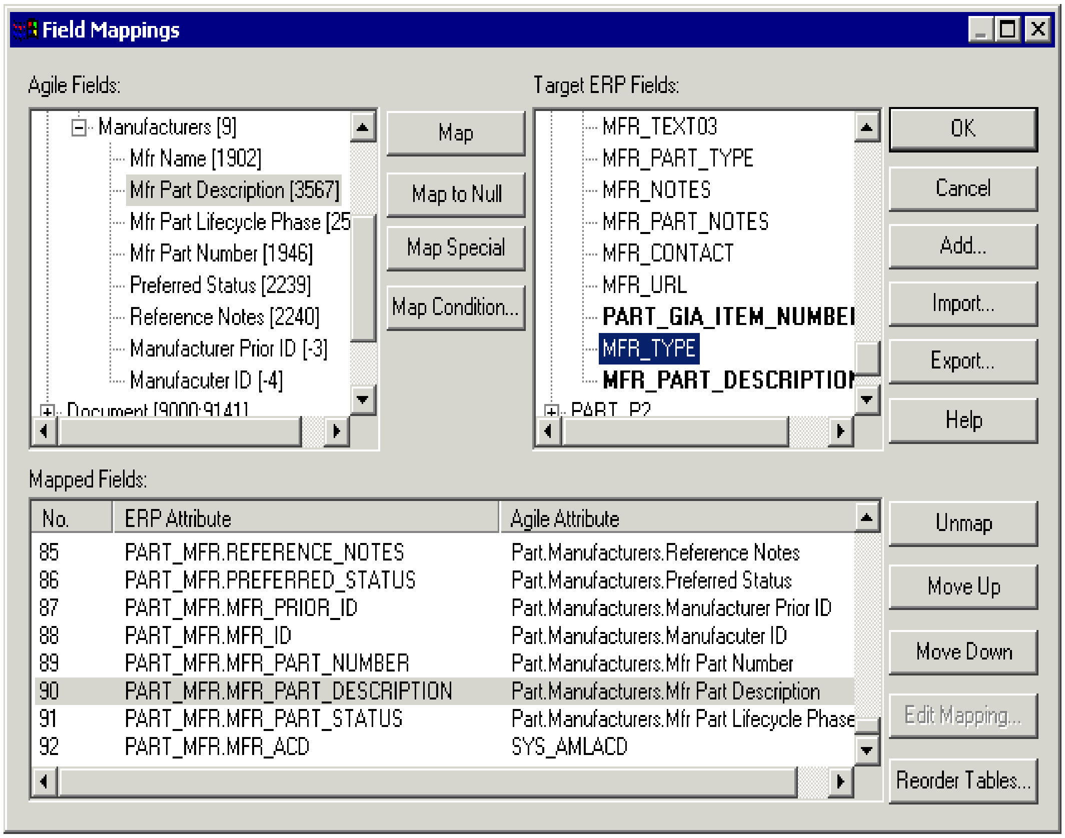
To map attributes to Agile attributes:

1. Choose **Mappings | Edit Mappings**..
2. Expand the AML branch in the **Target System Fields** tree.
3. Expand the Parts branch, then expand the Manufacturers branch. You find the manufacturer's

information.

Field Mapping dialog box

Equation 7: C-<>3: Field Mapping dialog box



No.	ERP Attribute	Agile Attribute
85	PART_MFR.REFERENCE_NOTES	Part.Manufacturers.Reference Notes
86	PART_MFR.PREFERRED_STATUS	Part.Manufacturers.Preferred Status
87	PART_MFR.MFR_PRIOR_ID	Part.Manufacturers.Manufacturer Prior ID
88	PART_MFR.MFR_ID	Part.Manufacturers.Manufacuter ID
89	PART_MFR.MFR_PART_NUMBER	Part.Manufacturers.Mfr Part Number
90	PART_MFR.MFR_PART_DESCRIPTION	Part.Manufacturers.Mfr Part Description
91	PART_MFR.MFR_PART_STATUS	Part.Manufacturers.Mfr Part Lifecycle Phase
92	PART_MFR.MFR_ACD	SYS_AMLACD

Note You need not map all the attributes that you define in the table.

File Format

To assure that the manufacturer part data (in this case the AML table) is formatted properly within the transfer file, ChangeCAST labels the Manufacturer Part table as a BOM table. You can see this in the **BOM/AML Table Settings** tab of the File Layout Settings dialog box (**Format | BOM/AML Table Settings**). This will generate the appropriate two-line logic required for the adapter to know what changes have occurred to which manufacturer for that item. In the following example, the manufacturer “ACE” is modified (“C”) from a status of Alternate to Preferred, and SPEEDY is added (“A”). The information is similar to that provided for a BOM.

```
# File Header Data

"MCO","M00003"

# AML Data

"GF-2000-02","ACE","ACE-CHASSIS-ASSEMBLY"," ","Alternate"," "," ","C"
```

```
"GF-2000-02","ACE","ACE-CHASSIS-ASSEMBLY"," ","Preferred"," "," ","A"
"GF-2000-02","SPEEDY","SPEEDY-CAR001"," ","Preferred"," "," ","A"
```

Application Adapters

Finally, the application adapter needs to be able to parse, validate, and update the target system with this new data. If installing or implementing an Agile AML-enabled adapter is an upgrade for you, make sure that complete regression testing is done on the entire adapter. Make sure that you can transmit simple part additions and multiple part additions, filter documents (if required), part revisions, created and modified BOM structures, and so on, as well as the manufacturer and manufacturer part data.

As noted earlier, the manufacturer's data is sent to the target system whenever a part is affected in a change order. Manufacturer's information will be sent in one of two ways, depending on the orientation of the BOM (Full vs. Delta). If a Full BOM is selected, all manufacturers for an affected item will be passed in the transfer file regardless of whether they have changed or not. If a Delta BOM is selected, only data for the manufacturers affected in the change will be passed. Regardless of whether the data for the Manufacturer Object has changed or not, it will be sent. No Add/Change/Delete/Unmodified record flag is available for the manufacturer object from ChangeCAST. The following MCO produces the transfer files below.

Full BOM Example

```
# AML Data
"GF-2000-02","ACE","ACE-CHASSIS-ASSEMBLY"," ","Preferred"," "," ","D"
"GF-2000-02","ACME","ACME-CHASSIS"," ","Alternate"," "," ","C"
"GF-2000-02","ACME","ACME-CHASSIS"," ","Preferred"," "," ","A"
"GF-2000-02","KST COMPANY","KST-ENGINES-3"," ","Alternate"," "," ","A"

"GF-2000-02","PYRAMID PARTS","PYR-CH1165"," ","Alternate"," "," ","U"
# MFR Data
"ACE","123 Any Street. ","California","95109","Approved"
"ACME"," "," "," ","Approved"
"KST COMPANY"," "," "," ","Approved"
"PYRAMID PARTS"," "," "," ","Approved"

# MFRPART Data
"ACE","ACE-CHASSIS-ASSEMBLY","ACTIVE","Manufacturer Part"
"ACME","ACME-CHASSIS","ACTIVE","Manufacturer Part"
"KST COMPANY","KST-ENGINES-3","ACTIVE","Manufacturer Part"
"PYRAMID PARTS","PYR-CH1165","ACTIVE","Manufacturer Part"
```

Delta BOM Example

```
# AML Data
"GF-2000-02","ACE","ACE-CHASSIS-ASSEMBLY"," ","Preferred"," "," ","D"
"GF-2000-02","ACME","ACME-CHASSIS"," ","Alternate"," "," ","C"
"GF-2000-02","ACME","ACME-CHASSIS"," ","Preferred"," "," ","A"
"GF-2000-02","KST COMPANY","KST-ENGINES-3"," ","Alternate"," "," ","A"


# MFR Data
"ACE","123 Any Street. ","California","95109","Approved"
"ACME"," "," "," ","Approved"
"KST COMPANY"," "," "," ","Approved"

# MFRPART Data
"ACE","ACE-CHASSIS-ASSEMBLY","ACTIVE","Manufacturer Part"
"ACME","ACME-CHASSIS","ACTIVE","Manufacturer Part"
"KST COMPANY","KST-ENGINES-3","ACTIVE","Manufacturer Part"
```

Summary

To maintain and transfer MCOs and other manufacturer information from Agile through ChangeCAST to the target system:

- Modify and make visible the appropriate Manufacturer fields in Agile Web Client.
- Define the appropriate target tables and fields and then map them to the appropriate Agile attributes in ChangeCAST, making sure that you select the AML table as a BOM table.
- Ensure that you enable the adapter to receive and manage this data.